

The Time Complexity of the Token Swapping Problem and Its Parallel Variants

Jun Kawahara¹, Toshiki Saitoh², and Ryo Yoshinaka³

¹ Graduate School of Information Science, NAIST, Japan

² Graduate School of Engineering, Kobe University, Japan

³ Graduate School of Information Sciences, Tohoku University, Japan

Abstract. The token swapping problem (TSP) and its colored version are reconfiguration problems on graphs. This paper is concerned with the complexity of the TSP and two new variants; namely parallel TSP and parallel colored TSP. For a given graph where each vertex has a unique token on it, the TSP requires to find a shortest way to modify a token placement into another by swapping tokens on adjacent vertices. In the colored version, vertices and tokens are colored and the goal is to relocate tokens so that each vertex has a token of the same color. Their parallel versions allow simultaneous swaps on non-incident edges in one step. We investigate the time complexity of several restricted cases of those problems and show when those problems become tractable and remain intractable.

1 Introduction

Yamanaka et al. [15] have introduced a kind of reconfiguration problem on graphs, called the *token swapping problem (TSP)*⁴. Suppose that we have a simple graph where each vertex is assigned a token. Each token is labeled with its unique goal vertex, which may be different from where the token is currently placed. We want to relocate every misplaced token to its goal vertex. What we can do is to swap the two tokens on the ends of an arbitrary edge. The problem is to decide how many swaps are needed to realize the goal token placement. The upper half of Figure 1 illustrates a problem instance and a solution. The graph has 4 vertices 1, 2, 3, 4 and 4 edges $\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}$. Each token i is initially put on the vertex $5 - i$. By swapping the tokens on the edges $\{3, 4\}, \{1, 3\}, \{2, 4\}, \{3, 4\}$ in this order, we can match the indices of the tokens and vertices.

Yamanaka et al. have presented several positive results on the TSP in addition to classical results which can be seen as special cases of the TSP [8]. Namely, graph classes for which the TSP can be solved in polynomial-time are paths, cycles, complete graphs and complete bipartite graphs. They showed that the TSP for general graphs belongs to NP. The NP-hardness is recently shown in the preliminary version [10] of this paper and by Miltzow et al. [11] and Bonnet et

⁴ No salesman is traveling in this paper.

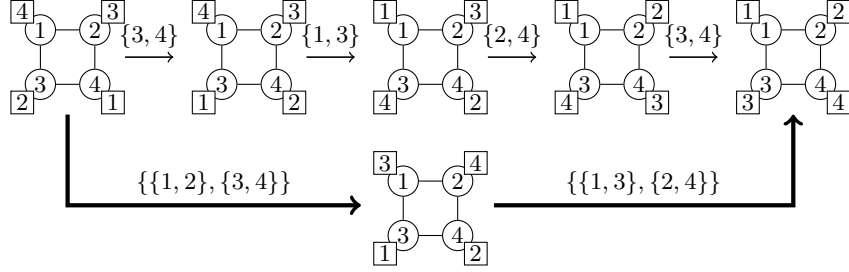


Fig. 1. Vertices and tokens are shown by circles and squares, respectively. Optimal solutions for the TSP and the PTSP are shown by small and big arrows, respectively.

al. [2] independently. On the other hand, some polynomial-time approximation algorithms are known for different classes of graphs including the general case [7, 11, 15]. For more backgrounds of the problem, the reader is referred to [15, 16].

A variant of the TSP is the *c-colored token swapping problem (c-CTSP)*. Tokens and vertices in the *c-CTSP* are colored by one of the *c* admissible colors. The *c-CTSP* is to decide how many swaps are required to relocate the tokens so that each vertex has a token of the same color. Yamanaka et al. [16] have investigated the *c-CTSP* and shown that the 3-CTSP is NP-complete while the 2-CTSP is solvable in polynomial time. This problem and a further generalization are also studied in [2].

This paper is concerned with the TSP and variants of it. First, we give a proof of the NP-hardness of the TSP.

- The TSP is NP-complete even when graphs are restricted to bipartite graphs where every vertex has degree at most 3 (Theorem 1).

In addition, we present two polynomial-time solvable subcases of the TSP. One is of lollipop graphs, which are combinations of a complete graph and a path. The other is the class of graphs which are combinations of a star and a path.

Variants of the TSP we will consider in this paper are the parallel versions of the TSP and *c-CTSP*. While in the TSP just one pair of tokens is swapped at once, the *parallel token swapping problem (PTSP)* allows us to swap token pairs on unadjacent edges simultaneously. We call a set of compatible swaps a *parallel swap*. The PTSP is to estimate how many parallel swaps are needed to achieve a goal token configuration. Figure 1 compares optimal solutions for the same instance of the TSP and the PTSP, where two parallel swaps are enough to relocate all the tokens to the goal vertices. Our main results concerning those problems include the following.

- The PTSP is NP-complete even to decide whether an instance admits a solution consisting of 3 parallel swaps (Theorem 4).
- One can decide in polynomial time whether an instance of the PTSP admits a solution consisting of 2 parallel swaps (Theorem 6).

- A polynomial-time algorithm that approximately solves the PTSP on paths is presented. It gives a parallel swap sequence whose length is at most one larger than that of an optimal solution (Theorem 7).
- The parallel 2-CTSP is NP-complete (Theorem 9).

The last result contrasts the fact that the 2-CTSP is solvable in polynomial-time [16].

One may consider the TSP and PTSP as special cases of the *minimum generator sequence problem (MGSP)* [5]. The MGSP is to determine whether one can obtain a permutation f on a finite set X by multiplying at most k permutations from a finite permutation set Π , where all of X , f , k and Π are input. The problem is known to be PSPACE-complete if k is specified in binary notation [8], while it becomes NP-complete if k is given in unary representation [5]. In the TSP and PTSP, permutation sets Π are restricted to the ones that have a graph representation. However, this does not necessarily mean that the NP-hardness of the PTSP implies the hardness of the MGSP, since the description size of all the admissible parallel swaps on a graph is exponential in the graph size.

2 Time Complexity of the Token Swapping Problem

We denote by $G = (V, E)$ an undirected graph whose vertex set is V and edge set is E . More precisely, elements of E are subsets of V consisting of exactly two distinct elements. A *configuration* f (on G) is a permutation on V , i.e., bijection from V to V . By $[u]_f$ we denote the orbit $\{f^i(u) \mid i \in \mathbb{N}\}$ of $u \in V$ under f . We call each element of V a *token* when we emphasize the fact that it is in the range of f . We say that a token v is on a vertex u in f if $v = f(u)$. A *swap* on G is a synonym for an edge of G , which behaves as a transposition. For a configuration f and a swap $e \in E$, the configuration obtained by applying e to f , which we denote by fe , is defined by

$$fe(u) = \begin{cases} f(v) & \text{if } e = \{u, v\}, \\ f(u) & \text{otherwise.} \end{cases}$$

For a sequence $\vec{e} = \langle e_1, \dots, e_m \rangle$ of swaps, the length m is denoted by $|\vec{e}|$. For $i \leq m$, by $\vec{e}_{\leq i}$ we denote the prefix $\langle e_1, \dots, e_i \rangle$. The configuration $f\vec{e}$ obtained by applying \vec{e} to f is $(\dots((fe_1)e_2)\dots)e_m$. We say that the token $f(u)$ on u is moved to v by \vec{e} if $f\vec{e}(v) = f(u)$. We count the total moves of each token $u \in V$ in the application as

$$\text{move}(f, \vec{e}, u) = |\{i \in \{1, \dots, m\} \mid (f\vec{e}_{\leq i-1})^{-1}(u) \neq (f\vec{e}_{\leq i})^{-1}(u)\}|.$$

Clearly $\text{move}(f, \vec{e}, u) \geq \text{dist}(f^{-1}(u), (f\vec{e})^{-1}(u))$, where $\text{dist}(u_1, u_2)$ denotes the length of a shortest path between u_1 and u_2 , and $\sum_{u \in V} \text{move}(f, \vec{e}, u) = 2|\vec{e}|$.

We denote the set of *solutions* for a configuration f by

$$\text{SOL}(G, f) = \{\vec{e} \mid \vec{e} \text{ is a swap sequence on } G \text{ such that } f\vec{e} \text{ is the identity}\}.$$

A solution $\vec{e}_0 \in \text{SOL}(G, f)$ is said to be *optimal* if $|\vec{e}_0| = \min\{|\vec{e}| \mid \vec{e} \in \text{SOL}(G, f)\}$. The length of an optimal solution is denoted by $\text{OPT}(G, f)$.

Problem 1 (Token Swapping Problem, TSP).

Instance: A connected graph G , a configuration f on G and a natural number k .

Question: $\text{OPT}(G, f) \leq k$?

2.1 TSP Is NP-complete

This subsection proves the NP-hardness of the TSP by a reduction from the 3DM, which is known to be NP-complete [9].

Problem 2 (Three dimensional matching problem, 3DM).

Instance: Three disjoint sets A_1, A_2, A_3 such that $|A_1| = |A_2| = |A_3|$ and a set $T \subseteq A_1 \times A_2 \times A_3$.

Question: Is there $M \subseteq T$ such that $|M| = |A_1|$ and every element of $A_1 \cup A_2 \cup A_3$ occurs just once in M ?

An instance of the 3DM is denoted by (A, T) where $A = A_1 \cup A_2 \cup A_3$ assuming that the partition is understood. Let $A_k = \{a_{k,1}, \dots, a_{k,n}\}$ for $k \in \{1, 2, 3\}$ and $T = \{t_1, \dots, t_m\}$. For notational convenience we write $a \in t$ if $a \in A$ occurs in $t \in T$ by identifying t with the set of the elements of t . We construct an instance (G_T, f) of the TSP as follows. The vertex set of G_T is $V_A \cup V_T$ with

$$\begin{aligned} V_A &= \{u_{k,i}, u'_{k,i} \mid k \in \{1, 2, 3\} \text{ and } i \in \{1, \dots, n\}\}, \\ V_T &= \{v_{j,k}, v'_{j,k} \mid j \in \{1, \dots, m\} \text{ and } k \in \{1, 2, 3\}\}. \end{aligned}$$

The edge set E_T is given by

$$\begin{aligned} E_T &= \{ \{u_{k,i}, v'_{j,k}\}, \{u'_{k,i}, v_{j,k}\} \mid a_{k,i} \in A_k \text{ occurs in } t_j \in T \} \\ &\cup \{ \{v_{j,k}, v'_{j,l}\} \subseteq V_T \mid j \in \{1, \dots, m\} \text{ and } k \neq l \}. \end{aligned}$$

We call the subgraph induced by $\{v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, v_{j,2}, v'_{j,3}\}$ the t_j -cycle. The initial configuration f is defined by

$$\begin{aligned} f(u_{k,i}) &= u'_{k,i} \text{ and } f(u'_{k,i}) = u_{k,i} \text{ for all } a_{k,i} \in A_k \text{ and } k \in \{1, 2, 3\}, \\ f(v_{j,k}) &= v_{j,k} \text{ and } f(v'_{j,k}) = v'_{j,k} \text{ for all } t_j \in T \text{ and } k \in \{1, 2, 3\}. \end{aligned}$$

In the initial configuration f , all and only the tokens in V_A are misplaced. Each token $u_{k,i} \in V_A$ on the vertex $u'_{k,i}$ must be moved to $u_{k,i}$ via (a part of) t_j -cycle for some $t_j \in T$ in which $a_{k,i}$ occurs. To design a short solution for (G_T, f) , it is desirable to have swaps at which both of the swapped tokens get closer to the destination. If (A, T) admits a solution, then one can find an optimal solution for (G_T, f) of length $21n$, where $9n$ of the swaps satisfy this property as we will see in Lemma 1. On the other hand, such an “efficient” solution is possible only when (A, T) admits a solution as shown in Lemma 2.

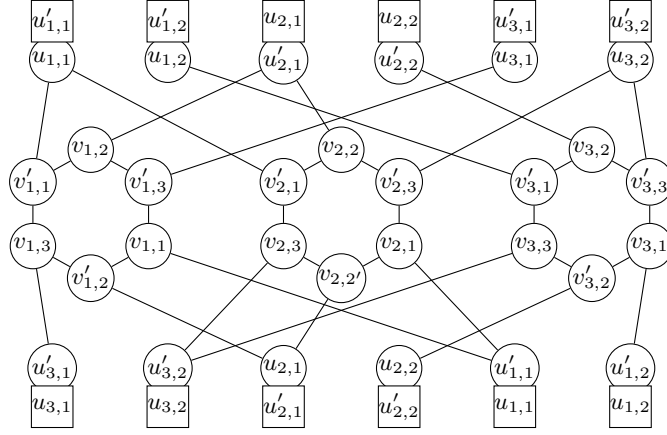


Fig. 2. The TSP instance reduced from the 3DM instance in Example 1. Vertices and tokens are denoted by circles and squares, respectively. The tokens which are already on the goal vertices in the initial configuration are omitted.

Example 1. Let $A = A_1 \cup A_2 \cup A_3$ and $T = \{t_1, t_2, t_3\}$ where $A_k = \{a_{k,1}, a_{k,2}\}$ for $k \in \{1, 2, 3\}$, $t_1 = \{a_{1,1}, a_{2,1}, a_{3,1}\}$, $t_2 = \{a_{1,1}, a_{2,1}, a_{3,2}\}$ and $t_3 = \{a_{1,2}, a_{2,2}, a_{3,2}\}$. Figure 2 shows the graph and initial configuration reduced from the 3DM instance (A, T) . This instance (A, T) has a solution $M = \{t_1, t_3\}$. The proof of Lemma 1 will give how to find an optimal solution for the reduced TSP instance corresponding to M . A part of the solution is illustrated in Figure 3.

Lemma 1. *If (A, T) has a solution then $\text{OPT}(G_T, f) \leq 21n$ with $n = |A_1|$.*

Proof. We show in the next paragraph that for each $t_j \in T$, there is a sequence σ_j of 21 swaps such that $g\sigma_j$ is identical to g except $(g\sigma_j)(u_{k,i}) = g(u'_{k,i})$ and $(g\sigma_j)(u'_{k,i}) = g(u_{k,i})$ if $a_{k,i}$ occurs in t_j for any configuration g . If $M \subseteq T$ is a solution, by collecting σ_j for all $t_j \in M$, we obtain a swap sequence σ_M of length $21n$ such that $f\sigma_M$ is the identity.

Let $t_j = (a_{1,i_1}, a_{2,i_2}, a_{3,i_3})$. We first move each of the tokens u_{k,i_k} on the vertex u'_{k,i_k} to the vertex $v_{j,k}$ and the tokens u'_{k,i_k} on u_{k,i_k} to $v'_{j,k}$. We then move the tokens u_{k,i_k} on $v_{j,k}$ to the opposite vertex $v'_{j,k}$ of the t_j -cycle for each $k \in \{1, 2, 3\}$ while moving u'_{k,i_k} on $v'_{j,k}$ to $v_{j,k}$ in the opposite direction simultaneously. At last we make swaps on the same 6 edges we used in the first phase. The above procedure consists of 21 swaps and gives the desired configuration. \square

Lemma 2. *If $\text{OPT}(G_T, f) \leq 21n$ with $n = |A_1|$ then (A, T) has a solution.*

Proof. We first show that $21n$ is a lower bound on $\text{OPT}(G_T, f)$. Suppose that $f\sigma$ is the identity. For each token $u_{k,i} \in V_A$, we have

$$\text{move}(f, \sigma, u_{k,i}) \geq \text{dist}(u_{k,i}, f^{-1}(u_{k,i})) = \text{dist}(u_{k,i}, u'_{k,i}) = 5.$$

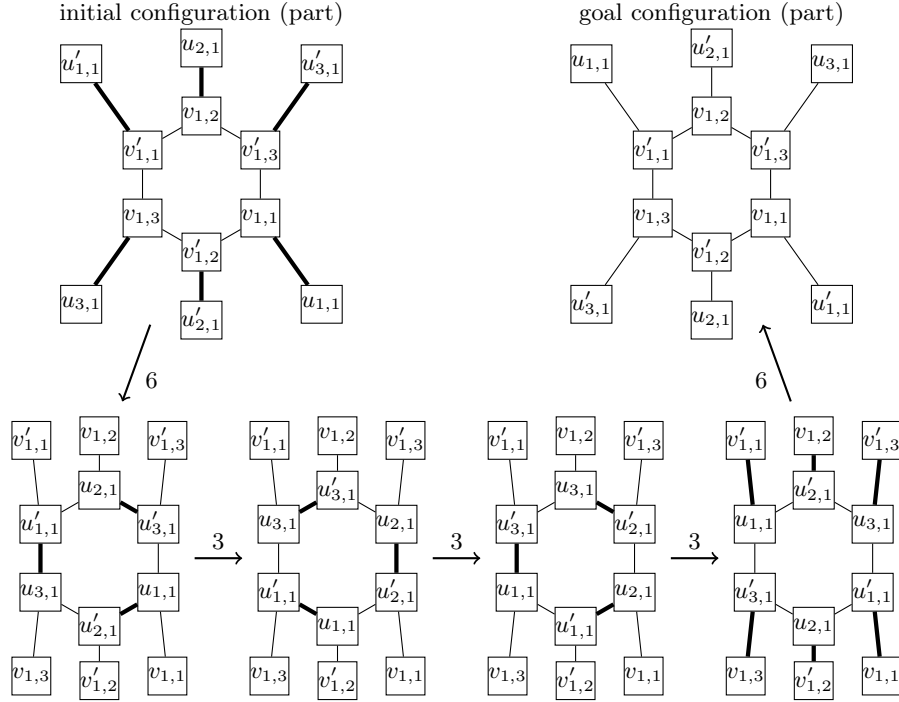


Fig. 3. The 3DM instance (A, T) of Example 1 has a solution $M = \{t_1, t_3\}$. The optimal solution given in the proof of Lemma 1 that exchanges $u_{k,1}$ and $u'_{k,1}$ for all $k \in \{1, 2, 3\}$ via the t_1 -cycle is illustrated here, where we suppress vertex names. By swapping the tokens on the bold edges in each configuration, we obtain the succeeding one pointed by an arrow. The number by each arrow shows the number of swaps. The swap sequence consists of 21 swaps in total. By doing the same on t_3 -cycle with respect to $u_{1,2}, u_{2,2}, u_{3,2}, u'_{1,2}, u'_{2,2}, u'_{3,2}$, we obtain the goal configuration.

The adjacent vertices of the vertex $u'_{k,i}$ are $v_{j,k}$ such that $a_{k,i} \in t_j$. Among those, let $\tau(u_{k,i}) \in V_T$ be the vertex to which $u_{k,i}$ goes for its first step, i.e., the first occurrence of $u'_{k,i}$ in σ is as $\{u'_{k,i}, \tau(u_{k,i})\}$. This means that $\text{move}(f, \sigma, \tau(u_{k,i})) \geq 2$, since the token $\tau(u_{k,i})$ must once leave from and later come back to the vertex $\tau(u_{k,i})$. The symmetric discussion holds for all tokens $u'_{k,i}$. Therefore, noting that τ is an injection, we obtain

$$|\sigma| = \frac{1}{2} \sum_{x \in V_A \cup V_T} \text{move}(f, \sigma, x) \geq \frac{1}{2} \sum_{x \in V_A} (\text{move}(f, \sigma, x) + \text{move}(f, \sigma, \tau(x))) \geq 21n.$$

This has shown that if $f\sigma$ is the identity and $|\sigma| \leq 21n$, then

- (1) $\text{move}(f, \sigma, x) = 5$ for all $x \in V_A$,
- (2) $\text{move}(f, \sigma, y) \neq 0$ for $y \in V_T$ if and only if $y = \tau(x)$ for some $x \in V_A$.

Let $M_\sigma = \{y \in V_T \mid \text{move}(f, \sigma, y) \neq 0\} = \{\tau(x) \in V_T \mid x \in V_A\}$. We are now going to prove that if $v_{j,1} \in M_\sigma$ then $\{v_{j,2}, v_{j,3}, v'_{j,1}, v'_{j,2}, v'_{j,3}\} \subseteq M_\sigma$, which implies that $\widetilde{M}_\sigma = \{t_j \in T \mid v_{j,1} \in M_\sigma\}$ is a solution for (A, T) .

Suppose $v_{j,1} \in M_\sigma$ and let $t_j \cap A_1 = \{a_{1,i}\}$. This means that $\tau(u_{1,i}) = v_{j,1}$ and $u_{1,i}$ goes from $u'_{1,i}$ to $u_{1,i}$ through $(u'_{1,i}, v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, u_{1,i})$ or $(u'_{1,i}, v_{j,1}, v'_{j,3}, v_{j,2}, v'_{j,1}, u_{1,i})$ by (2) and (1). In either case, $v'_{j,1} \in M_\sigma$. Suppose that $u_{1,i}$ takes the former $(u'_{1,i}, v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, u_{1,i})$. Then $v'_{j,2}, v_{j,3} \in M_\sigma$. Just like $v_{j,1} \in M_\sigma$ implies $v'_{j,1} \in M_\sigma$, we now see $v_{j,2}, v'_{j,3} \in M_\sigma$. \square

It is known that the 3DM is still NP-complete if each $a \in A$ occurs at most three times in T [6]. Assuming that T satisfies this constraint, it is easy to see that G_T is a bipartite graph with maximum vertex degree 3.

Theorem 1. *The TSP is NP-complete even on bipartite graphs with maximum vertex degree 3.*

2.2 PTIME Subcases of TSP

In this subsection, we present two graph classes on which the TSP can be solved in polynomial time. One is that of *lollipop graphs*, which are obtained by connecting a path and a complete graph with a bridge. That is, a lollipop graph is $L_{m,n} = (V, E)$ where $V = \{-m, \dots, -1, 0, 1, \dots, n\}$ and

$$E = \{\{i, j\} \subseteq V \mid i < j \leq 0 \text{ or } j = i + 1 > 0\}.$$

The other class consists of graphs obtained by connecting a path and a star. A *star-path graph* is $Q_{m,n} = (V, E)$ such that $V = \{-m, \dots, -1, 0, 1, \dots, n\}$ and

$$E = \{\{i, 0\} \subseteq V \mid i < 0\} \cup \{\{i, i+1\} \subseteq V \mid i \geq 0\}.$$

Algorithms 1 and 2 give optimal solutions for the TSP on lollipop and star-path graphs in polynomial time, respectively. Proofs are found in Appendices A and B.

Algorithm 1 TSP Algorithm for Lollipop Graphs

Input: A lollipop graph $L_{m,n}$ and a configuration f on $L_{m,n}$
for $k = n, \dots, 1, 0, -1, \dots, -m$ **do**
 Move the token k to the vertex k directly;
end for

Algorithm 2 TSP Algorithm for Star-Path Graphs

Input: A star-path graph $Q_{m,n}$ and a configuration f on $Q_{m,n}$
for $k = n, \dots, 1, 0, -1, \dots, -m$ **do**
 while the token on the vertex 0 has an index less than 0 **do**
 Move the token on the vertex 0 to its goal vertex;
 end while
 Move the token k to the vertex k ;
end for

3 Parallel Token Swapping Problem

The *Parallel token swapping problem (PTSP)* is the parallel version of the TSP. Definitions and notation for the TSP are straightforwardly generalized for the PTSP. A *parallel swap* S on G is a synonym for an involution which is a subset of E , or for a matching of G , i.e., $S \subseteq E$ such that $\{u, v_1\}, \{u, v_2\} \in S$ implies $v_1 = v_2$. For a configuration f and a parallel swap $S \subseteq E$, the configuration obtained by applying S to f is defined by $fS(u) = f(v)$ if $\{u, v\} \in S$ and $fS(u) = f(u)$ if $u \notin \bigcup S$. Let

$$\text{P-SOL}(G, f) = \{ \vec{S} \mid \vec{S} \text{ is a parallel swap sequence s.t. } f\vec{S} \text{ is the identity} \}$$

$$\text{P-OPT}(G, f) = \min \{ |\vec{S}| \mid \vec{S} \in \text{P-SOL}(G, f) \}.$$

Problem 3 (Parallel Token Swapping Problem, PTSP).

Instance: A connected graph G , a configuration f on G and a natural number k .

Question: $\text{P-OPT}(G, f) \leq k$?

It is trivial that $\text{P-OPT}(G, f) \leq \text{OPT}(G, f) \leq \text{P-OPT}(G, f)|V|/2$, since any parallel swap S consists of at most $|V|/2$ (single) swaps. Since $\text{OPT}(G, f) \leq |V|(|V| - 1)/2$ holds [15], the PTSP belongs to NP.

Yamanaka et al. [15] discussed the relation between the TSP and parallel sorting on an SIMD machine consisting of several processors with local memory which are connected by a network [1]. The relation to the PTSP is more direct.

Theorem 2. *If there is a parallel sorting algorithm with r rounds for an inter-connection network G , then $\text{P-OPT}(G, f) \leq r$ for any configuration f on G .*

3.1 PTSP Is NP-complete

We show the NP-hardness of the PTSP by a reduction from a restricted kind of the satisfiability problem, which we call *PPN-Separable 3SAT* (*Sep-SAT* for short). For a set X of (*Boolean*) *variables*, $\neg X$ denotes the set of their negative literals. A *3-clause* is a subset of $X \cup \neg X$ whose cardinality is at most 3. An instance of the Sep-SAT is a finite collection F of 3-clauses, which can be partitioned into three subsets $F_1, F_2, F_3 \subseteq F$ such that for each variable $x \in X$, the positive literal x occurs just once in each of F_1, F_2 and the negative literal $\neg x$ occurs just once in F_3 .

Theorem 3. *The Sep-SAT is NP-complete.*

Proof. See Appendix C. □

We give a reduction from the Sep-SAT to the PTSP. For a given instance $F = \{C_1, \dots, C_n\}$ over a variable set $X = \{x_1, \dots, x_m\}$ of the Sep-SAT, we define a graph $G_F = (V_F, E_F)$ in the following manner. Let F be partitioned into F_1, F_2, F_3 where each of F_1 and F_2 has just one occurrence of each variable as a positive literal and F_3 has just one occurrence of each negative literal. Define

$$V_F = \{u_i, u'_i, u_{i,1}, u_{i,2}, u_{i,3}, u_{i,4} \mid 1 \leq i \leq m\} \\ \cup \{v_j, v'_j \mid 1 \leq j \leq n\} \cup \{v_{j,i} \mid x_i \in C_j \text{ or } \neg x_i \in C_j\}.$$

The edge set E_F is the least set that makes G_F contain the following paths of length 3:

$$(u_i, u_{i,1}, u_{i,2}, u'_i) \text{ and } (u_i, u_{i,3}, u_{i,4}, u'_i) \text{ for each } i \in \{1, \dots, m\}, \\ (v_j, v_{j,i}, u_{i,k}, v'_j) \text{ if } x_i \in C_j \in F_k \text{ or } \neg x_i \in C_j \in F_k.$$

It is not hard to see that G_F is a bipartite graph. Vertices v_j and v'_j have degree at most 3 for $j \in \{1, \dots, n\}$, while $u_{i,k}$ has degree 4 for $i \in \{1, \dots, m\}$ and $k \in \{1, 2, 3\}$. The initial configuration f is defined to be the identity except

$$f(u_i) = u'_i, \quad f(u'_i) = u_i, \quad f(v_j) = v'_j, \quad f(v'_j) = v_j,$$

for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$.

Example 2. For $X = \{x_1, x_2, x_3\}$, let F consist of $C_1 = \{x_1, x_2\}$, $C_2 = \{x_3\}$, $C_3 = \{x_1\}$, $C_4 = \{x_2, x_3\}$ and $C_5 = \{\neg x_1, \neg x_2, \neg x_3\}$. Then F is partitioned into $F_1 = \{C_1, C_2\}$, $F_2 = \{C_3, C_4\}$ and $F_3 = \{C_5\}$, where each variable occurs just once in each F_k with $k \in \{1, 2, 3\}$. Moreover, F_1 and F_2 have only positive literals and F_3 has only negative literals. Therefore, F is a Sep-SAT instance. Figure 4 shows the reduction from F . The formula F is satisfied by assigning 1 to x_1, x_3 and 0 to x_2 . Corresponding to this assignment, by moving misplaced tokens along the bold edges in Figure 4, the goal configuration is realized in 3 steps.

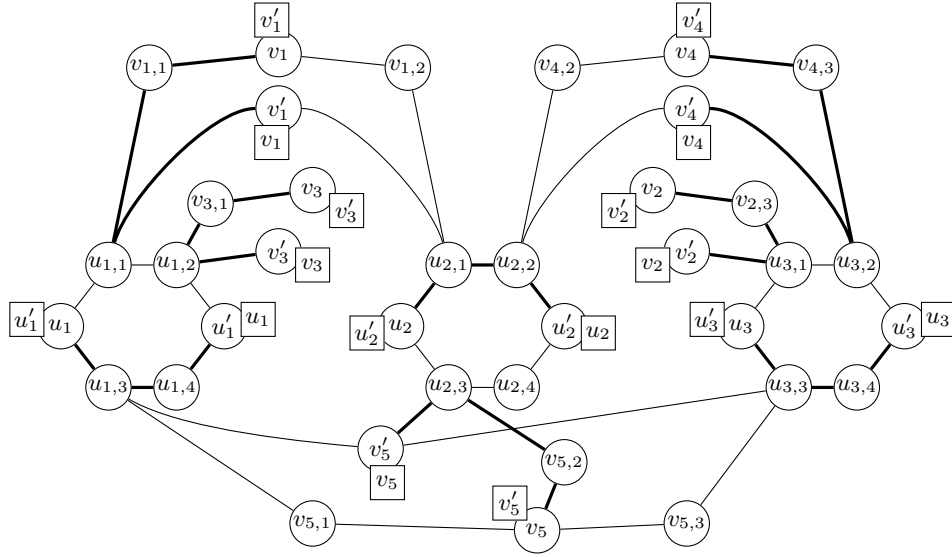


Fig. 4. The TSP instance obtained from the Sep-SAT instance F of Example 2. By moving misplaced tokens along the bold edges, the goal configuration is realized in 3 steps. The reduction graph described in the proof for Theorem 5 has essentially the same shape.

Since $\text{dist}(w, f(w)) = 3$ if $w \neq f(w)$, obviously $\text{P-OPT}(G_F, f) \geq 3$. We will show that F is satisfiable if and only if this lower bound is achieved. Here we describe an intuition behind the reduction by giving the following observation between a 3-step solution for (G_F, f) and a solution for F :

- tokens u_i and u'_i pass vertices $u_{i,1}$ and $u_{i,2}$ iff x_i should be assigned 0, while they pass over $u_{i,3}$ and $u_{i,4}$ iff x_i should be assigned 1,
- if tokens v_j and v'_j pass a vertex $u_{i,k}$ for some $k \in \{1, 2\}$ then $C_j \in F_k$ is satisfied thanks to x_i , while if they pass over $u_{i,3}$ then $C_j \in F_3$ is satisfied thanks to $\neg x_i$.

Of course it is contradictory that a clause $C_j \in F_1$ is satisfied by $x_i \in C_j$ which is assigned 0. This impossibility corresponds to the fact that there are no i, j such that both u_i and v_j with $C_j \in F_1$ go to their respective goals via $u_{i,1}$ in a 3-step solution.

Theorem 4. *To decide whether $\text{P-OPT}(G, f) \leq 3$ is NP-complete even when G is restricted to be a bipartite graph with maximum vertex degree 4.*

Proof. We show that F is satisfiable if and only if $\text{P-OPT}(G_F, f) = 3$.

Suppose that there is $\phi : X \rightarrow \{0, 1\}$ satisfying F . Then each clause must have a literal to which ϕ assigns 1. Let $\psi : F \rightarrow X$ be such that $\psi(C_j) \in C_j$ and

$\phi(\psi(C_j)) = 1$ if $C_j \in F_1 \cup F_2$, and $\neg\psi(C_j) \in C_j$ and $\phi(\psi(C_j)) = 0$ if $C_j \in F_3$.
Define

$$\begin{aligned} S_1 &= \{ \{u_i, u_{i,1}\}, \{u'_i, u_{i,2}\} \mid \phi(x_i) = 0 \} \cup \{ \{u_i, u_{i,3}\}, \{u'_i, u_{i,4}\} \mid \phi(x_i) = 1 \} \\ &\quad \cup \{ \{v_j, v_{j,i}\}, \{v'_j, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \}, \\ S_2 &= \{ \{u_{i,1}, u_{i,2}\} \mid \phi(x_i) = 0 \} \cup \{ \{u_{i,3}, u_{i,4}\} \mid \phi(x_i) = 1 \} \\ &\quad \cup \{ \{v_{j,i}, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \}. \end{aligned}$$

It is not hard to see that $\langle S_1, S_2, S_1 \rangle$ is a solution for (G_F, f) .

Conversely, suppose that (G_F, f) admits a solution $\langle S_1, S_2, S_3 \rangle$. Since the token on u_i is moved to u'_i by the three steps, the path that u'_i takes should be either $\langle u_i, u_{i,1}, u_{i,2}, u'_i \rangle$ or $\langle u_i, u_{i,3}, u_{i,4}, u'_i \rangle$. In other words, S_2 contains at least one of $\{u_{i,1}, u_{i,2}\}$ and $\{u_{i,3}, u_{i,4}\}$. We prove that F is satisfied by the assignment $\phi : X \rightarrow \{0, 1\}$ defined as

$$\phi(x_i) = \begin{cases} 0 & \text{if } \{u_{i,1}, u_{i,2}\} \in S_2, \\ 1 & \text{otherwise.} \end{cases}$$

For each $C_j \in F_1$, the token on v_j must be moved to v'_j via $u_{i,1}$ for some i such that $x_i \in C_j$. That is, $\{v_{j,i}, u_{i,1}\} \in S_2$. Since S_2 is a parallel swap, $\{u_{i,1}, u_{i,2}\} \notin S_2$ in this case, which means $\phi(x_i) = 1$. Hence C_j is satisfied by ϕ . Almost the same arguments show that clauses in F_2 and F_3 are also satisfied by ϕ . \square

One can modify the above reduction so that every vertex has degree at most 3 by dividing vertices $u_{i,k}$ into two vertices of degree at most 3. Let

$$\begin{aligned} V_F &= \{ u_i, u'_i, u_{i,1}, u'_{i,1}, u_{i,2}, u'_{i,2}, u_{i,3}, u'_{i,3}, u_{i,4}, u'_{i,4} \mid 1 \leq i \leq m \} \\ &\quad \cup \{ v_j, v'_j \mid 1 \leq j \leq n \} \cup \{ v_{j,i}, v'_{j,i} \mid x_i \in C_j \text{ or } \neg x_i \in C_j \}. \end{aligned}$$

Our graph G_F contains the following paths of length 5:

$$\begin{aligned} & (u_i, u_{i,1}, u'_{i,1}, u_{i,2}, u'_{i,2}, u'_i) \text{ and } (u_i, u_{i,3}, u'_{i,3}, u_{i,4}, u'_{i,4}, u'_i) \text{ for each } i \in \{1, \dots, m\}, \\ & (v_j, v_{j,i}, u_{i,k}, u'_{i,k}, v'_{j,i}, v'_j) \text{ if } x_i \in C_j \in F_k \text{ or } \neg x_i \in C_j \in F_k. \end{aligned}$$

The initial configuration f is defined in the same manner as the previous construction. It is identity except $f(u_i) = u'_i$, $f(u'_i) = u_i$, $f(v_j) = v'_j$, and $f(v'_j) = v_j$ for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$.

Theorem 5. *To decide whether $\text{P-OPT}(G, f) \leq 5$ is NP-complete even when G is restricted to be a bipartite graph with maximum vertex degree 3.*

3.2 PTIME Subcases of PTSP

In this subsection we discuss tractable subcases of the PTSP. In contrast to Theorem 4, the 2-step PTSP is decidable in polynomial time. In addition, we present an approximation algorithm for finding a solution for the PTSP on paths whose length can be at most one larger than that of an optimal solution.

2-Step PTSP It is well-known that any permutation can be expressed as a product of 2 involutions, which means that any problem instance of the PTSP on a complete graph has a 2-step solution. Graphs we treat are not necessarily complete but the arguments by Petersen and Tenner [13] on involution factorization lead to the following observation, which is useful to decide whether $\text{P-OPT}(G, f) \leq 2$ for general graphs G .

Proposition 1. $\langle S, T \rangle \in \text{P-SOL}(G, f)$ if and only if the set of orbits under f is partitioned as $\{\{[u_1]_f, [v_1]_f\}, \dots, \{[u_k]_f, [v_k]_f\}\}$ (possibly $[u_j]_f = [v_j]_f$ for some $j \in \{1, \dots, k\}$) so that for every $j \in \{1, \dots, k\}$,

$$\{f^i(u_j), f^{-i}(v_j)\} \in \tilde{S} \text{ and } \{f^{i+1}(u_j), f^{-i}(v_j)\} \in \tilde{T} \text{ for all } i \in \mathbb{Z},$$

where $\tilde{S} = S \cup \{\{v\} \mid v \in V - \bigcup S\}$ for a parallel swap S .

Theorem 6. It is decidable in polynomial time if $\text{P-OPT}(G, f) \leq 2$ for any G and f .

Proof. Suppose G and f are given. One can compute in polynomial time all the orbits $[\cdot]_f$. Let us denote the subgraph of G induced by a vertex set $U \subseteq V$ by G_U and the sub-configuration of f restricted to $[u]_f \cup [v]_f$ by $f_{u,v}$. The set

$$\Gamma_f = \{\{[u]_f, [v]_f\} \mid \text{P-OPT}(G_{[u]_f \cup [v]_f}, f_{u,v}) \leq 2\}$$

can be computed in polynomial time by Proposition 1. It is clear that $\text{P-OPT}(G, f) \leq 2$ if and only if there is a subset $\Gamma \subseteq \Gamma_f$ in which every orbit occurs exactly once. This problem is a very minor variant of the problem of finding a perfect matching on a graph, which can be solved in polynomial time [4]. \square

One can calculate the number of 2-step solutions in $\text{P-SOL}(K_n, f)$ for any configuration on the complete graph K_n using Petersen and Tenner's formula [13]. On the other hand, it is a $\#P$ -complete problem to calculate $|\text{P-SOL}(G, f)|$ for general graphs G . This can be shown by a reduction from the problem of calculating the number of perfect matchings in a bipartite graph, which is known to be $\#P$ -complete [14]. For $H = (V, E)$, let the vertex set of G be $V' = \{u_i \mid u \in V \text{ and } i \in \{1, 2\}\}$ and the edge set $E' = \{(u_i, v_j) \mid (u, v) \in E \text{ and } i, j \in \{1, 2\}\}$. The initial configuration is defined by $f(u_1) = u_2$ and $f(u_2) = u_1$ for all $u \in V$. Then it is easy to see that $|\text{P-SOL}(G, f)| = 2^m$ for the number m of perfect matchings in H . Note that if H is bipartite, then so is G .

Approximation Algorithm for the PTSP on Paths We present an approximation algorithm for the PTSP on paths which outputs a parallel swap sequence whose length is no more than $\text{P-OPT}(P_n, f) + 1$, where $P_n = (\{1, \dots, n\}, \{\{i, i+1\} \mid 1 \leq i < n\})$ and f is a configuration on P_n . We say that a swap $\{i, i+1\} \in E$ is *reasonable w.r.t. f* if $f(i) > f(i+1)$, and moreover, a parallel swap sequence $\tilde{S} = \langle S_1, \dots, S_m \rangle$ is *reasonable w.r.t. f* if every $e \in S_j$ is reasonable w.r.t. $f \langle S_1, \dots, S_{j-1} \rangle$ for all $j \in \{1, \dots, m\}$. The parallel swap sequence $\langle S_1, \dots, S_m \rangle$

output by Algorithm 3 is reasonable and satisfies the condition which we call the *odd-even condition*: for each odd number j , all swaps in S_j are of the form $\{2i-1, 2i\}$ for some $i \geq 1$, and for each even number j , all swaps in S_j are of the form $\{2i, 2i+1\}$ for some $i \geq 1$. Our algorithm computes a reasonable odd-even parallel swap sequence in a greedy manner.

Lemma 3. *Suppose that $g = fS$ for a reasonable parallel swap S w.r.t. f . For any $\langle S_1, \dots, S_m \rangle \in \text{P-SOL}(P_n, f)$, there is $\langle S'_1, \dots, S'_m \rangle \in \text{P-SOL}(P_n, g)$ such that $S'_j \subseteq S_j$ for all $j \in \{1, \dots, m\}$.*

The lemma implies that we may assume without loss of generality that an optimal solution $\langle S_1, \dots, S_m \rangle$ satisfies the following conditions:

- it is reasonable,
- if $f\langle S_1, \dots, S_j \rangle(i) > f\langle S_1, \dots, S_j \rangle(i+1)$ then $\{i, i+1\} \cap \bigcup S_{j+1} \neq \emptyset$ for $j < m$.

Algorithm 3 Approximation algorithm for PTSP on paths

Input: A configuration f_0 on P_n

Output: A solution $\vec{S} \in \text{P-SOL}(P_n, f_0)$

Let $j = 0$;

while f_j is not identity **do**

Let $j = j + 1$, $S_j = \{ \{i, i+1\} \mid f_{j-1}(i) > f_{j-1}(i+1) \text{ and } i+j \text{ is even} \}$ and $f_j = f_{j-1}S_j$;

end while

return $\langle S_1, \dots, S_j \rangle$;

Let us denote the output of Algorithm 3 by $\text{AP}(P_n, f_0)$.

Theorem 7. $\text{AP}(P_n, f_0) \in \text{P-SOL}(P_n, f_0)$ and $|\text{AP}(P_n, f_0)| \leq \text{P-OPT}(P_n, f_0) + 1$.

Proof. Let $\vec{T} = \text{AP}(P_n, f_0)$. It is obvious that $\vec{T} \in \text{P-SOL}(P_n, f_0)$ and it is odd-even. It is easy to see by Lemma 3 that $|\vec{T}| \leq |\vec{S}|$ for any odd-even solution $\vec{S} \in \text{P-SOL}(P_n, f_0)$.

We next show that every swap sequence $\vec{S} = \langle S_1, \dots, S_m \rangle$ admits an equivalent odd-even sequence that is not much longer than the original. Without loss of generality we assume that $S_j \cap S_{j+1} = \emptyset$ for any j (in fact, any reasonable parallel swap sequence meets this condition). For a parallel swap sequence $\vec{S} = \langle S_1, \dots, S_m \rangle$, define $\mathfrak{CE}(\vec{S}) = \langle S'_1, \dots, S'_{m+1} \rangle$ by delaying swaps which do not meet the odd-even condition, that is,

$$S'_j = \{ \{i, i+1\} \in S_j \cup S_{j-1} \mid i+j \text{ is even} \}$$

for $j = 1, \dots, m+1$ assuming that $S_0 = S_{m+1} = \emptyset$. By the parity restriction, each S'_j is a parallel swap. It is easy to show by induction on j that

$$f\langle S'_1, \dots, S'_j \rangle(i) = \begin{cases} f\langle S_1, \dots, S_{j-1} \rangle(i) & \text{if } \{i, i+1\} \in S_j \text{ and } i+j \text{ is odd,} \\ f\langle S_1, \dots, S_j \rangle(i) & \text{otherwise,} \end{cases}$$

for each $j \in \{1, \dots, m+1\}$, which implies that $f\vec{S} = f\mathfrak{C}(\vec{S})$. Therefore, for an optimal reasonable solution \vec{S}_0 , we have $|\vec{S}_0| + 1 = |\mathfrak{C}(\vec{S}_0)| \geq |\vec{T}|$. \square

4 Parallel Colored Token Swapping Problem

The *colored token swapping problem* (CTSP) is a generalization of the TSP, where each token is colored and different tokens may have the same color. By swapping tokens on adjacent vertices, the goal coloring configuration should be realized. More formally, a *coloring* is a map f from V to \mathbb{N} . The definition of a swap application to a configuration can be applied to colorings with no change. We say that two colorings f and g are *consistent* if $|f^{-1}(i)| = |g^{-1}(i)|$ for all $i \in \mathbb{N}$. Since the problem is a generalization of the TSP, obviously it is NP-hard. Yamanaka et al. [16] have investigated subcases of the CTSP called the c -CTSP where the codomain of colorings is restricted to $\{1, \dots, c\}$. We discuss the parallel version of the c -CTSP in this section.

Problem 4 (Parallel c -Colored Token Swapping Problem, c -PCTSP).

Instance: A graph G , two consistent c -colorings f and g , and a number $k \in \mathbb{N}$.

Question: Is there \vec{S} with $|\vec{S}| \leq k$ such that $f\vec{S} = g$?

Define $\text{P-OPT}(G, f, g) = \min\{|\vec{S}| \mid f\vec{S} = g\}$ for two consistent colorings f and g . Since $\text{P-OPT}(G, f, g)$ can be bounded by $\text{P-OPT}(G, h)$ for some configuration h , the c -PTSP belongs to NP.

Yamanaka et al. have shown that the 3-CTSP is NP-hard by a reduction from the 3DM. It is not hard to see that their reduction works to prove the NP-hardness of the 3-PCTSP. We then obtain the following theorem as a corollary to their discussion.

Theorem 8. *To decide whether $\text{P-OPT}(G, f, g) \leq 3$ is NP-hard even if G is restricted to be a planar bipartite graph with maximum vertex degree 3 and f and g are 3-colorings.*

Yamanaka et al. have shown that the 2-CTSP is solvable in polynomial time on the other hand. In contrast, we prove that the 2-PCTSP is still NP-hard.

Theorem 9. *To decide whether $\text{P-OPT}(G, f, g) \leq 3$ is NP-hard for a bipartite graph G with maximum vertex degree 4 and 2-colorings f and g .*

Proof. We prove the theorem by a reduction from the Sep-SAT. We use the same graph used in the proof of Theorem 4. The initial and goal colorings f and g are defined to be $f(w) = 1$ and $g(w) = 1$ for all w but $f(u_i) = g(u'_i) = 2$ for each $x_i \in X$, $f(v_j) = g(v'_j) = 2$ for each $C_j \in F_1 \cup F_3$ and $f(v'_j) = g(v_j) = 2$ for each $C_j \in F_2$. The claim that F is satisfiable if and only if $\text{P-OPT}(G_F, f, g) = 3$ can be established by the same manner as the proof of Theorem 4. \square

We can also show the following using the ideas for proving Theorems 5 and 8.

Theorem 10. *To decide whether $\text{P-OPT}(G, f, g) \leq 5$ is NP-hard even if G is a bipartite graph with maximum vertex degree 3 and f and g are 2-colorings.*

References

- [1] D. Bitton, D. J. DeWitt, D. K. Hsiao, and J. Menon. A taxonomy of parallel sorting. *ACM Comput. Surv.*, 16(3):287–318, 1984.
- [2] É. Bonnet, T. Miltzow, and P. Rzażewski. Complexity of Token Swapping and its Variants. *CoRR*, abs/1607.07676, 2016.
- [3] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. of the 3rd Annual ACM Symposium on Theory of Computing*, pp. 151–158. ACM, 1971.
- [4] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.
- [5] S. Even and O. Goldreich. The minimum-length generator sequence problem is NP-hard. *J. Algorithms*, 2(3):311–313, 1981.
- [6] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [7] L. S. Heath and J. P. C. Vergara. Sorting by short swaps. *J. Comput. Biol.*, 10(5):775–789, 2003.
- [8] M. Jerrum. The complexity of finding minimum-length generator sequences. *Theor. Comput. Sci.*, 36:265–289, 1985.
- [9] R. M. Karp. Reducibility among combinatorial problems. In *the Complexity of Computer Computations*, The IBM Research Symposia Series, pp. 85–103. Springer, 1972.
- [10] J. Kawahara, T. Saitoh, and R. Yoshinaka. A note on the complexity of the token swapping problem (in Japanese). *IPSJ SIG Technical Report*, 2015-AL-156(3):1–7, January 2016.
- [11] T. Miltzow, L. Narrins, Y. Okamoto, G. Rote, A. Thomas, and T. Uno. Approximation and Hardness of Token Swapping. *ESA*, 66:1–66:15, 2016.
- [12] I. Pak. Reduced decompositions of permutations in terms of star transpositions, generalized catalan numbers and k -ARY trees. *Discrete Math.*, 204(1-3):329–335, 1999.
- [13] T. K. Petersen and B. E. Tenner. How to write a permutation as a product of involutions (and why you might care), *Integers.*, 13:#A63, 2013.
- [14] L. G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8, pp. 189–201. 1979.
- [15] K. Yamanaka, E. D. Demaine, T. Ito, J. Kawahara, M. Kiyomi, Y. Okamoto, T. Saitoh, A. Suzuki, K. Uchizawa, and T. Uno. Swapping labeled tokens on graphs. In *Fun with Algorithms*, LNCS 8496, pp. 364–375. Springer, 2014.
- [16] K. Yamanaka, T. Horiyama, D. G. Kirkpatrick, Y. Otachi, T. Saitoh, R. Uehara, and Y. Uno. Swapping colored tokens on graphs. In *Algorithms and Data Structures*, LNCS 9214, pp. 619–628. Springer, 2015.

A Proof that the TSP on Lollipop Graphs Is in P

This appendix gives a proof that Algorithm 1 computes an optimal solution for the TSP on lollipop graphs. We will give an evaluation function on configurations on lollipop graphs $L_{m,n}$ such that any swap changes the value by one, every swap by the algorithm reduces the value by one, and the value is 0 if and only if the configuration is the identity. Algorithm 1 first moves non-negative tokens to the goal vertices on the path and then moves negative ones in the clique. The number of swaps needed to move a token $j \in \{0, \dots, n\}$ is evaluated by

$$\pi(f, j) = \begin{cases} j + 1 & \text{if } f^{-1}(j) < 0, \\ \min(j + 1, \text{Inv}(f, j)) & \text{if } f^{-1}(j) \geq 0, \end{cases}$$

where

$$\text{Inv}(f, j) = |\{i \mid i < j \text{ and } f^{-1}(i) > f^{-1}(j)\}|.$$

So it takes

$$\pi(f) = \sum_{j=0}^n \pi(f, j)$$

swaps to move the non-negative tokens to the goal vertices in total. We then move the negative tokens in the clique. For a configuration f' such that $f'(j) = j$ for all $j \geq 0$, the number of swaps needed is

$$\nu(f') = m - |\Lambda_{f'}| \text{ where } \Lambda_{f'} = \{[i]_{f'} \mid i < 0\}. \text{ (See e.g. [8])}$$

We need to evaluate $|\Lambda_{f'}|$ for $f' = f\vec{S}$ where \vec{S} moves all the non-negative tokens to their goals. Let us call an injection f from $\{-m, \dots, k\}$ to $\{-m, \dots, n\}$ for some $k \in \{-1, 0, \dots, n\}$ a *pseudo configuration* if the range of f includes $\{-m, \dots, -1\}$. For notational simplicity, a pseudo configuration f will often be identified with the sequence $\langle f(-1), \dots, f(-m), f(0), \dots, f(k) \rangle$ or the sequence pair $(\langle f(-1), \dots, f(-m) \rangle; \langle f(0), \dots, f(k) \rangle)$. Let $\vec{i} = \langle f(-1), \dots, f(-m) \rangle$, $\vec{j} = \langle f(0), f(1), \dots, f(k) \rangle$ and $c = \max(\vec{i}) \geq 0$. We define ν recursively on $|\vec{j}|$ by

$$\nu(\vec{i}; \vec{j}) = \begin{cases} m - |\Lambda_f| & \text{if } \vec{j} \text{ is empty,} \\ \nu(\vec{i}; \langle f(1), \dots, f(k) \rangle) & \text{if } c < f(0), \\ \nu(\vec{i}[f(0)/c]; \langle f(1), \dots, f(k) \rangle) & \text{if } c > f(0), \end{cases}$$

where $[a/b]$ replaces b by a . That is,

$$f[a/b](i) = \begin{cases} f(i) & \text{if } f(i) \neq b, \\ a & \text{if } f(i) = b. \end{cases}$$

Note that $(\vec{i}[f(0)/c]; \langle f(1), \dots, f(k) \rangle)$ above is a pseudo configuration. Our evaluation function Φ is given as

$$\Phi(f) = \pi(f) + \nu(f).$$

It is clear that $\Phi(f) \geq 0$ for any configuration and the equation holds if and only if f is the identity.

Lemma 4. For any \vec{i}, \vec{j} , there is a sequence \vec{i}' consisting of the m smallest elements from $\vec{i} \cdot \vec{j}$, where $\vec{i} \cdot \vec{j}$ denotes the concatenation of \vec{i} and \vec{j} , such that for any \vec{k}

$$\nu(\vec{i}; \vec{j} \cdot \vec{k}) = \nu(\vec{i}'; \vec{k}),$$

provided that $(\vec{i}; \vec{j} \cdot \vec{k})$ is a pseudo configuration.

Lemma 5. If $\vec{i} \cdot \vec{j}_1$ contains m or more tokens smaller than $a \geq 0$, then

$$\nu(\vec{i}; \vec{j}_1 \cdot a \cdot \vec{j}_2) = \nu(\vec{i}; \vec{j}_1 \cdot \vec{j}_2),$$

provided that $(\vec{i}; \vec{j}_1 \cdot a \cdot \vec{j}_2)$ is a pseudo configuration.

Now we are going to prove that any possible swap on the graph changes the value of Φ by one. We have three cases depending on where the swap takes place.

Lemma 6. Let $f = (\vec{i}; \vec{j})$ and $g = (\vec{i}'; \vec{j})$ be pseudo configurations such that $\vec{i}' = \vec{i}[a/b, b/a]$ for some distinct tokens a, b . Then

$$|\nu(f) - \nu(g)| = 1.$$

Proof. We show this by induction on the definition of ν . If \vec{j} is not empty, the claim follows the induction hypothesis immediately. If \vec{j} is empty, f and g are configurations on the clique of $\{-1, \dots, -m\}$.

Case 1. Suppose $[a]_f = [b]_f$. Let $k = |[a]_f|$ and $b = f^j(a)$. Then $g^i(a) = f^i(a)$ for $i < j$, $g^j(a) = a$, $g^i(b) = f^{j+i}(a)$ for $i < k - j$ and $g^{k-j}(b) = b$. That is, $[a]_f = [a]_g \cup [b]_g$, $[a]_g \neq [b]_g$ and $|A_g| = |A_f| + 1$. Hence $\nu(g) = \nu(f) - 1$.

Case 2. Suppose $[a]_f \neq [b]_f$. Let $k_a = |[a]_f|$ and $k_b = |[b]_f|$. Then $g^i(a) = f^i(a)$ for $i \in \{0, \dots, k_a - 1\}$, $g^{k_a+i}(a) = f^i(b)$ for $i \in \{0, \dots, k_b - 1\}$ and $g^{k_a+k_b}(a) = a$. That is, $[a]_g = [b]_g = [a]_f \cup [b]_f$ and $|A_g| = |A_f| - 1$. Hence $\nu(g) = \nu(f) + 1$. \square

Corollary 1. Suppose that $g = fe$ for some swap $e \subseteq \{-1, \dots, -m\}$. Then $|\Phi(g) - \Phi(f)| = 1$.

Proof. Clearly $\pi(f) = \pi(g)$ by definition. The claim follows Lemma 6. \square

Lemma 7. If $g = f\{h, 0\}$ with $h < 0$, then $|\Phi(g) - \Phi(f)| = 1$.

Proof. We may assume without loss of generality that $h = -1$ for the symmetry. Let

$$\begin{aligned} \nu(f) &= \nu(a \cdot \vec{i}; b \cdot \vec{j}), \\ \nu(g) &= \nu(b \cdot \vec{i}; a \cdot \vec{j}). \end{aligned}$$

Without loss of generality we assume $a > b$.

Case 1. Suppose $a, b < 0$. Clearly $\pi(f) = \pi(g)$ and $\max(\vec{i}) \geq 0$. We have $|\nu(f) - \nu(g)| = 1$ by applying Lemma 6 to the fact

$$\begin{aligned} \nu(f) &= \nu(a \cdot \vec{i}[b/c]; \vec{j}), \\ \nu(g) &= \nu(b \cdot \vec{i}[a/c]; \vec{j}), \end{aligned}$$

where $c = \max(\vec{i})$.

Case 2. Suppose $a \geq 0 > b$.

Case 2.1. Suppose $a > \max(\vec{i})$. We have

$$\nu(f) = \nu(g) = \nu(b \cdot \vec{i}; \vec{j}),$$

All the m elements of $b \cdot \vec{i}$ are smaller than a , which are among $m + a$ tokens smaller than a . Therefore, \vec{j} contains exactly a tokens smaller than a , which means $\pi(g, a) = a$. On the other hand, $\pi(f, k) = \pi(g, k)$ for all other positive tokens k but $\pi(f, a) = a + 1$ by definition.

All in all, $\Phi(f) - \Phi(g) = 1$.

Case 2.2. Suppose $\max(\vec{i}) > a$. Let $c = \max(\vec{i})$. We have

$$\nu(f) = \nu(a \cdot \vec{i}[b/c]; \vec{j}),$$

$$\nu(g) = \nu(b \cdot \vec{i}[a/c]; \vec{j}),$$

and $|\nu(f) - \nu(g)| = 1$ by Lemma 6.

The fact $c > a$ implies at most $m - 1$ tokens in $b \cdot \vec{i}$ are smaller than a , which are among $m + a$ tokens smaller than a . Hence \vec{j} contains at least $a + 1$ tokens smaller than a , which means $\pi(g, a) = a + 1$. Therefore, $\pi(f, k) = \pi(g, k)$ for all other positive tokens k .

All in all, $|\Phi(f) - \Phi(g)| = 1$.

Case 3. Suppose $a > b \geq 0$. This case is almost identical to Case 2 except that we need to confirm $\pi(f, b) = \pi(g, b)$ in addition. The fact $a > b$ implies at most $m - 1$ tokens in $a \cdot \vec{i}$ are smaller than b , and \vec{j} contains at least $b + 1$ tokens smaller than b , which means $\pi(f, b) = \pi(g, b) = b + 1$. \square

Lemma 8. If $g = f\{k, k + 1\}$ for some $k \geq 0$, then $|\Phi(g) - \Phi(f)| = 1$.

Proof. Let

$$f = (\vec{i}; \vec{j}_1 \cdot \langle a, b \rangle \cdot \vec{j}_2),$$

$$g = (\vec{i}; \vec{j}_1 \cdot \langle b, a \rangle \cdot \vec{j}_2).$$

By Lemma 4, there exists \vec{i}' consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\nu(f) = \nu(\vec{i}'; \langle a, b \rangle \cdot \vec{j}_2),$$

$$\nu(g) = \nu(\vec{i}'; \langle b, a \rangle \cdot \vec{j}_2).$$

Without loss of generality we assume $a > b$.

Case 1. Suppose $a, b < 0$. Clearly $\pi(f) = \pi(g)$. For the two largest tokens c and d in \vec{i}' with $c > d$, we have

$$\nu(f) = \nu(\vec{i}'[a/c, b/d]; \vec{j}_2),$$

$$\nu(g) = \nu(\vec{i}'[b/c, a/d]; \vec{j}_2).$$

Lemma 6 implies $|\Phi(f) - \Phi(g)| = 1$.

Case 2. Suppose $a \geq 0$. We have $\text{Inv}(f, a) = \text{Inv}(g, a) + 1$.

Case 2.1. Suppose $\text{Inv}(f, a) \leq a$. In this case, we have $\pi(g, a) = \text{Inv}(g, a) = \text{Inv}(f, a) - 1 = \pi(f, a) - 1$ and thus $\pi(f) = \pi(g) + 1$. The fact that $b \cdot \vec{j}_2$ contains at most a tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains at least m tokens smaller than a . That is, all of \vec{i}' are smaller than a . By Lemma 5, we have

$$\nu(f) = \nu(g) = \nu(\vec{i}'; b \cdot \vec{j}_2).$$

All in all, $\Phi(f) = \Phi(g) + 1$.

Case 2.2. Suppose $\text{Inv}(f, a) = a + 1$. In this case, we have $\pi(f, a) = a + 1$, $\pi(g, a) = \text{Inv}(g, a) = a$ and thus $\pi(f) = \pi(g) + 1$. The fact that $b \cdot \vec{j}_2$ contains exactly $a + 1$ tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains exactly $m - 1$ tokens smaller than a . That is, all of \vec{i}' are smaller than a except one token $c = \max(\vec{i}')$. Therefore,

$$\begin{aligned}\nu(f) &= \nu(\vec{i}'[a/c]; b \cdot \vec{j}_2) = \nu(\vec{i}'[b/c]; \vec{j}_2), \\ \nu(g) &= \nu(\vec{i}'[b/c]; a \cdot \vec{j}_2) = \nu(\vec{i}'[b/c]; \vec{j}_2).\end{aligned}$$

All in all, $\Phi(f) = \Phi(g) + 1$.

Case 2.3. Suppose $\text{Inv}(f, a) > a + 1$. In this case, we have $\pi(f, a) = \pi(g, a) = a + 1$ and $\pi(f) = \pi(g)$. The fact that $b \cdot \vec{j}_2$ contains at least $a + 2$ tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains at most $m - 2$ tokens smaller than a . That is, the two largest tokens c and d in \vec{i}' are bigger than a . Therefore,

$$\begin{aligned}\nu(f) &= \nu(\vec{i}'[a/c, b/d]; \vec{j}_2), \\ \nu(g) &= \nu(\vec{i}'[b/c, a/d]; \vec{j}_2).\end{aligned}$$

Lemma 6 implies $|\nu(f) - \nu(g)| = 1$. All in all, $|\Phi(f) - \Phi(g)| = 1$. □

Corollary 2. $\Phi(f) \leq \text{OPT}(L_{m,n}, f)$.

Proof. By Corollary 1 and Lemmas 7 and 8. □

Lemma 9. Suppose that our algorithm changes f to g at a point in the run. Then $\Phi(g) = \Phi(f) - 1$.

Proof. Suppose that the algorithm moves a token $a \geq 0$. If $f^{-1}(a) < 0$ then Case 2.1 of the proof of Lemma 7 applies and we have $\Phi(f) = \Phi(g) + 1$. If $f^{-1}(a) \geq 0$, the fact that $f(i) < a$ for all $i < 0$ implies that $\text{Inv}(f, a) \leq a$. Hence Case 2.1 of the proof of Lemma 8 applies and we have $\Phi(f) = \Phi(g) + 1$.

Suppose that the algorithm moves a token $a < 0$. Then Case 1 of Lemma 6 applies. We conclude $\Phi(f) = \Phi(g) + 1$. □

Therefore, our algorithm gives a solution of $\Phi(f)$ steps, which is optimal by Corollary 2.

Theorem 11. The TSP on lollipop graphs can be solved in polynomial time.

B Proof that the TSP on Star-Path Graphs Is in P

This appendix gives a proof that Algorithm 2 computes an optimal solution for the TSP on star-path graphs $Q_{m,n}$ in a manner similar to Appendix A. The number of swaps needed to move non-negative tokens to the goal vertices is evaluated by the same function π . On the other hand, the number of swaps needed to relocate negative tokens is evaluated differently from the case of lollipop graphs. The negative tokens which must be moved are in $N_f = \{ f(i) \in \{-m, \dots, -1\} \mid f(i) \neq i \}$. Among those, some are on a non-negative vertex and some are on a negative vertex. Tokens of the former type will be forced to move to 0 by the moves of non-negative tokens (Type B) and then go to the goal vertex (Type A). Moves of Type B are counted by π . On the other hand, tokens i of the latter type form equivalence classes $[i]_f \subseteq N_f$, which require $[i]_f + 1$ swaps to be relocated to the goal vertices. Let

$$\Delta_f = \{ [i]_f \subseteq N_f \mid i < 0 \}$$

and

$$\mu(f) = |N_f| + |\Delta_f|.$$

It is easy to see that $\mu(f)$ correctly evaluates the number of swaps required to relocate negative tokens in the star graph [12, 15]. One might think $\pi(f) + \mu(f)$ could be the right evaluation for $\text{OPT}(Q_{m,n}, f)$. However, when the vertex 0 is occupied by a negative token $i < 0$ and the vertex i is occupied by the positive token j which is the largest among the tokens on negative vertices, then the move of i to i (Type A) causes the right move of j to 0, which reduces the number of swaps required to move j to the goal. That is, actually π overestimates the number of swaps for j . We must discount the evaluation from $\pi(f) + \mu(f)$. For a pseudo configuration $f = (\vec{i}; \vec{j}) = (\langle i_1, \dots, i_m \rangle; \langle j_1, \dots, j_k \rangle)$ and $c = \max(\vec{i})$, define

$$\delta(\vec{i}; \vec{j}) = \begin{cases} 0 & \text{if } c < 0, \\ \delta(\vec{i}; \langle j_2, \dots, j_k \rangle) & \text{if } j_1 > c \geq 0, \\ \delta(\vec{i}[j_1/c]; \langle j_2, \dots, j_k \rangle) & \text{if } c \geq j_1 \geq 0, \\ \delta(\vec{i}[j_1/i_{-j_1}]; \vec{j}[i_{-j_1}/j_1]) - 1 & \text{if } j_1 < 0 \text{ and } i_{-j_1} = c \geq 0, \\ \delta(\vec{i}[j_1/i_{-j_1}]; \vec{j}[i_{-j_1}/j_1]) & \text{otherwise.} \end{cases}$$

Our evaluation function Ψ is given as

$$\Psi(f) = \pi(f) + \mu(f) + \delta(f).$$

It is clear that $\Psi(f) \geq 0$ for any configuration f and $\Psi(f) = 0$ if f is the identity.

For a pseudo configuration $(\langle i_1, \dots, i_m \rangle; a)$, let us define

$$\gamma(\langle i_1, \dots, i_m \rangle; a) = \begin{cases} \gamma(\langle i_1, \dots, i_{-a-1}, a, i_{-a+1}, \dots, i_m \rangle; i_{-a}) & \text{if } a < 0, \\ (\langle i_1, \dots, i_m \rangle; a) & \text{if } a \geq 0. \end{cases}$$

The function γ simulates the **while** loop of Algorithm 2 in the sense that if the algorithm has $(\vec{i}; a \cdot \vec{j})$ as the value of f at the beginning of the **while** loop, it will be $(\vec{i}'; a' \cdot \vec{j})$ when exiting the loop for $(\vec{i}'; a') = \gamma(\vec{i}; a)$.

Lemma 10. *Let $\gamma(\vec{i}; a) = (\vec{i}'; b)$. Then*

$$\delta(\vec{i}; a \cdot \vec{j}) = \begin{cases} \delta(\vec{i}'; b \cdot \vec{j}) - 1 & \text{if } b = \max(\vec{i}), \\ \delta(\vec{i}'; b \cdot \vec{j}) & \text{otherwise.} \end{cases}$$

Lemma 11. *For any \vec{i}, \vec{j} , there are an integer $\alpha \leq 0$ and a sequence \vec{i}' consisting of the m smallest elements from $\vec{i} \cdot \vec{j}$ such that for any \vec{k}*

$$\delta(\vec{i}; \vec{j} \cdot \vec{k}) = \delta(\vec{i}'; \vec{k}) + \alpha$$

provided that $(\vec{i}; \vec{j} \cdot \vec{k})$ is a pseudo configuration.

Lemma 12. *If $\vec{i} \cdot \vec{j}_1$ contains m or more tokens smaller than $k \geq 0$, then*

$$\delta(\vec{i}; \vec{j}_1 \cdot k \cdot \vec{j}_2) = \delta(\vec{i}; \vec{j}_1 \cdot \vec{j}_2)$$

provided that $(\vec{i}; \vec{j}_1 \cdot k \cdot \vec{j}_2)$ is a pseudo configuration.

Proof. By induction on $|\vec{j}_1|$. □

Corollary 3. *For any configuration $f = \langle \vec{i}; \vec{j}_1 \cdot k \cdot \vec{j}_2 \rangle$ with $k \geq 0$, if $\text{Inv}(f, k) \leq k$ then*

$$\delta(\vec{i}; \vec{j}_1 \cdot k \cdot \vec{j}_2) = \delta(\vec{i}; \vec{j}_1 \cdot \vec{j}_2).$$

In particular if \vec{i} contains negative tokens only, $\delta(f) = 0$.

Proof. Recall that there exist just $k+m$ tokens smaller than k . The fact $\text{Inv}(f, k) \leq k$ means that \vec{j}_2 contains at most k tokens smaller than k , so $\vec{i} \cdot \vec{j}_1$ must have at least k such tokens. □

B.1 Ψ Evaluates Our Algorithm

Lemma 13. *Suppose that our algorithm changes f to g at a point in the run. Then $\Psi(g) = \Psi(f) - 1$.*

Proof. We have two types of swaps.

Case A. When the algorithm moves the token $f(0) < 0$ to the vertex $f(0)$ (Type A). In this case we have $|N_g| = |N_f| - 1$. Let $a = f(0)$ and $b = f(a)$, which implies $g(0) = b$ and $g(a) = a$. Let $I = \{f(-1), \dots, f(-m)\}$ be the set of tokens on the negative vertices in f .

Case A.1. $0 \leq b = \max I$. Clearly $\delta(f) = \delta(g) - 1$ and $\mu(f) = \mu(g) + 1$. Since $\pi(f, i) = \pi(g, i)$ for all $i \neq b$, it is enough to show $\pi(f, b) = \pi(g, b) + 1$. By definition $\pi(f, b) = b + 1$. Recall that there are exactly $b + m$ tokens that are smaller than b . Since the m tokens on the negative vertices in g are all smaller

than b , there are exactly b tokens smaller than b under g . That is, $\text{Inv}(g, b) = b$ and thus $\pi(g, b) = b = \pi(f, b) - 1$.

Case A.2. $0 \leq b < \max I$. Clearly $\delta(f) = \delta(g)$ and $\mu(f) = \mu(g) + 1$. To see $\pi(f, i) = \pi(g, i)$ for all $i \in \{0, \dots, n\} - \{b\}$ is trivial, so it is enough to show $\pi(f, b) = \pi(g, b)$. By definition $\pi(f, b) = b + 1$. Recall that there are exactly $b + m$ tokens that are smaller than b , of which at most $m - 1$ tokens can be on negative vertices in g , since at least one negative vertex is occupied by a token bigger than b . Therefore, there are at least $b + 1$ tokens smaller than b on non-negative vertices in g . That is, $\text{Inv}(g, b) \geq b + 1$ and thus $\pi(g, b) = b + 1 = \pi(f, b)$.

Case A.3. $b < 0$. Clearly $\pi(f, i) = \pi(g, i)$ for all $i \geq 0$ and $\delta(f) = \delta(g)$ by definition. One can easily see $\Delta_f = \Delta_g$, for $[a]_f = [b]_f \notin \Delta_f$, $[a]_g \notin \Delta_g$ and $[b]_b \notin \Delta_g$. Hence $\mu(g) = \mu(f) - 1$.

Case B. When the algorithm moves a token k as a move of Type B.

Case B.1. $k \geq 0$ and $f^{-1}(k) < 0$. Let $a = f^{-1}(k)$ and $f(0) = b$, that is, $g(a) = b$ and $g(0) = k$. By the behavior of the algorithm, we have $f(i) \leq k$ for all $i \leq 0$. Since $b \geq 0$, we have $\mu(f) = \mu(g)$ and $\delta(f) = \delta(g)$. It is trivially true that $\pi(f, i) = \pi(g, i)$ for all $i \in \{0, \dots, n\} - \{k, b\}$. Thus it is enough to show that $\pi(g, k) + \pi(g, b) = \pi(f, k) + \pi(f, b) - 1$. By definition $\pi(f, k) = k + 1$ and $\pi(g, b) = b + 1$. Since all the m tokens on the negative vertices in g are smaller than k , the other k tokens smaller than k are found on some non-negative vertices. That is, $\text{Inv}(g, k) = k$ and thus $\pi(g, k) = k = \pi(f, k) - 1$. On the other hand in f , at least one token, namely k , on a negative vertex is bigger than b . Therefore, at least $b + 1$ tokens smaller than b are on some non-negative vertices in f . That is, $\text{Inv}(f, b) \geq b + 1$ and thus $\pi(f, b) = b + 1$. Therefore, $\pi(g) = \pi(f) - 1$.

Case B.2. $k \geq 0$ and $f^{-1}(k) \geq 0$. Clearly $\mu(g) = \mu(f)$, $\text{Inv}(g, k) = \text{Inv}(f, k) - 1$ and $\text{Inv}(g, j) = \text{Inv}(f, j)$ for all $j \in \{0, \dots, n\} - \{k\}$. By the behavior of the algorithm, $f(j) = j$ for all $j > k$ and thus $\text{Inv}(f, k) \leq k$ and $\pi(g, k) = \pi(f, k) - 1$. Hence $\pi(g) = \pi(f) - 1$. Corollary 3 implies $\delta(g) = \delta(f)$.

Case B.3. $k < 0$. The case where $f^{-1}(k) = 0$ can be discussed as in Case A.3. We assume $f^{-1}(k) < 0$, in which case we have $f(i) = i$ for all $i \geq 0$ by the behavior of the algorithm. Clearly $[k]_f \in \Delta_f$ and $\Delta_g = \Delta_f - \{[k]_f\}$, thus $|\Delta_g| = |\Delta_f| - 1$ and $\mu(g) = \mu(f) - 1$. On the other hand, $\pi(f, 0) = 0$ and $\pi(g, 0) = 1$, while $\pi(f, j) = \pi(g, j)$ for all $j > 0$. We have $\delta(f) = 0$ by Corollary 3 and $\delta(g) = -1$ by Lemma 10. \square

B.2 Ψ Is the Right Evaluation Function

Now we are going to prove that any possible swap on the graph changes the value of Ψ by one. We have 6 cases depending on the signs of swapped tokens and the vertices where the swap takes place. Namely we discuss cases where the tokens are both non-negative (Lemma 16), where one is non-negative and the other is negative (Lemma 17) and where both are negative (Lemma 18). Each case has two subcases depending on whether one of the tokens is on a negative vertex or not. Lemmas 14 and 15 are useful to prove those lemmas.

Lemma 14. Let $(\vec{i}; \vec{j})$ and $(\vec{i}'; \vec{j})$ be pseudo configurations such that \vec{i} contains two distinct non-negative numbers $a, b \geq 0$ and $\vec{i}' = \vec{i}[a/b, b/a]$. Then

$$|\delta(\vec{i}; \vec{j}) - \delta(\vec{i}'; \vec{j})| = 1.$$

Note that \vec{j} cannot be empty, since $\vec{i} \cdot \vec{j}$ is a pseudo configuration.

Proof. It is enough to show that for any $a, b \geq 0$, \vec{i}, \vec{j} and d ,

$$|\delta(\langle a, b \rangle \cdot \vec{i}; d \cdot \vec{j}) - \delta(\langle b, a \rangle \cdot \vec{i}; d \cdot \vec{j})| = 1.$$

We show this claim by induction on the definition of δ . If $d \notin \{-1, -2\}$, the proof is trivial. For the symmetry, we discuss the case where $d = -1$ only. Without loss of generality we assume $a < b$. Let $c = \max(\vec{i})$.

Case 1. In the case where $b > c$, we have

$$\begin{aligned} \delta(\langle a, b \rangle \cdot \vec{i}; -1 \cdot \vec{j}) &= \delta(\langle -1, b \rangle \cdot \vec{i}; a \cdot \vec{j}) = \delta(\langle -1, a \rangle \cdot \vec{i}; \vec{j}), \\ \delta(\langle b, a \rangle \cdot \vec{i}; -1 \cdot \vec{j}) &= \delta(\langle -1, a \rangle \cdot \vec{i}; \vec{j}) - 1. \end{aligned}$$

The claim holds.

Case 2. In the case where $b < c$,

$$\begin{aligned} \delta(\langle a, b \rangle \cdot \vec{i}; -1 \cdot \vec{j}) &= \delta(\langle -1, b \rangle \cdot \vec{i}; a \cdot \vec{j}) = \delta(\langle -1, b \rangle \cdot \vec{i}[a/c]; \vec{j}), \\ \delta(\langle b, a \rangle \cdot \vec{i}; -1 \cdot \vec{j}) &= \delta(\langle -1, a \rangle \cdot \vec{i}; b \cdot \vec{j}) = \delta(\langle -1, a \rangle \cdot \vec{i}[b/c]; \vec{j}). \end{aligned}$$

The claim follows the induction hypothesis. \square

Let $f = (\vec{i}; \vec{j})$ be a pseudo configuration where \vec{i} contains a negative token a . The a -resolution of \vec{i} is defined by

$$\vec{i}' = \vec{i}[a/f(a), f(a)/f^2(a), \dots, f^{k-1}(a)/f^k(a), f^k(a)/a]$$

where k is the least natural number such that either $f^{k+1}(a) = a$ or $f^k(a) \geq 0$. That is, we relocate tokens $a, f(a), \dots, f^{k-1}(a)$ on negative vertices to their respective goals and push $f^k(a)$ out to a , which is actually its goal if $f^{k+1}(a) = a$. We also call $g = (\vec{i}'; \vec{j})$ the a -resolution of f . If $\gamma(\vec{i}; a) = (\vec{j}; b)$ and $a < 0$, it is easy to see that \vec{j} is the a -resolutions of $\vec{i}[a/b]$.

Lemma 15. If g is the a -resolution of f , then $\delta(f) = \delta(g)$.

Proof. By induction on the definition of δ . \square

Lemma 16. Suppose that g is obtained from f by swapping non-negative tokens. Then $|\Psi(g) - \Psi(f)| = 1$.

Proof. Suppose that non-negative tokens a and b are swapped. Obviously $\mu(g) = \mu(f)$. We have two cases depending on where those tokens are swapped.

Case 1. The swap takes place on two non-negative vertices. That is,

$$\begin{aligned} f &= (\vec{i}; \vec{j}_1 \cdot \langle a, b \rangle \cdot \vec{j}_2), \\ g &= (\vec{i}; \vec{j}_1 \cdot \langle b, a \rangle \cdot \vec{j}_2), \end{aligned}$$

for some $a, b \geq 0$. Without loss of generality we assume $a < b$. We have $\pi(f, i) = \pi(g, i)$ for all $i \in \{0, \dots, n\} - \{b\}$, while $\text{Inv}(g, b) = \text{Inv}(f, b) + 1$. By Lemma 11, there exists \vec{i}' consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\begin{aligned}\delta(f) &= \delta(\vec{i}'; \langle a, b \rangle \cdot \vec{j}_2) + \alpha, \\ \delta(g) &= \delta(\vec{i}'; \langle b, a \rangle \cdot \vec{j}_2) + \alpha\end{aligned}$$

for some $\alpha \leq 0$.

Case 1.1. $\text{Inv}(f, b) < b$ and $\text{Inv}(g, b) < b + 1$. In this case, we have $\pi(f, b) = \text{Inv}(f, b)$, $\pi(g, b) = \text{Inv}(g, b)$ and thus $\pi(g) = \pi(f) + 1$. Corollary 3 applies to both f and g and we obtain

$$\delta(g) = \delta(f) = \delta(\vec{i}; \vec{j}_1 \cdot a \cdot \vec{j}_2)$$

and $\Psi(g) = \Psi(f) + 1$.

Case 1.2. $\text{Inv}(f, b) = b$ and $\text{Inv}(g, b) = b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. Since \vec{j}_2 contains b tokens smaller than b , on the other hand $\vec{i} \cdot \vec{j}_1$ contains exactly $m - 1$ tokens smaller than b . Let c be the unique element of \vec{i}' which is bigger than b . Then

$$\begin{aligned}\delta(f) &= \delta(\vec{i}'[a/c]; b \cdot \vec{j}) + \alpha = \delta(\vec{i}'[a/c]; \vec{j}) + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c]; a \cdot \vec{j}) + \alpha = \delta(\vec{i}'[a/c]; \vec{j}) + \alpha,\end{aligned}$$

since b is the biggest in $\vec{i}'[b/c]$. Hence $\delta(g) = \delta(f)$ and $\Psi(g) = \Psi(f) + 1$.

Case 1.3. $\text{Inv}(f, b) > b$ and $\text{Inv}(g, b) > b + 1$. In this case, $\pi(f, b) = \pi(g, b) = b + 1$ and $\pi(g) = \pi(f)$. Since \vec{j}_2 contains at least $b + 1$ tokens smaller than b , $\vec{i} \cdot \vec{j}_1$ contains at most $m - 2$ tokens smaller than b . Hence \vec{i}' contains at least 2 tokens bigger than b . Let c_1 and c_2 be the biggest and second biggest in \vec{i}' , respectively. Then

$$\begin{aligned}\delta(f) &= \delta(\vec{i}'[a/c_1, b/c_2]; b \cdot \vec{j}) + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c_1, a/c_2]; a \cdot \vec{j}) + \alpha.\end{aligned}$$

By Lemma 14, we obtain $|\delta(g) - \delta(f)| = 1$ and $|\Psi(g) - \Psi(f)| = 1$.

Case 2. The swap takes place between a negative vertex and 0. Without loss of generality we may assume that the negative position is -1 and $f(-1) < g(-1)$. That is,

$$\begin{aligned}f &= (a \cdot \vec{i}; b \cdot \vec{j}), \\ g &= (b \cdot \vec{i}; a \cdot \vec{j}),\end{aligned}$$

where $0 \leq a < b$. By definition $\pi(f, a) = a + 1$ and $\pi(g, b) = b + 1$. Since there are $a + m$ tokens smaller than a , of which at most $m - 1$ tokens can be in $b \cdot \vec{i}$, we have $\text{Inv}(g, a) \geq a + 1$. That is, $\pi(g, a) = \pi(f, a)$. On the other hand, since there are $b + m$ tokens smaller than b , of which at most m tokens can be in $a \cdot \vec{i}$, we have $\text{Inv}(f, b) \geq b$.

Case 2.1. $\text{Inv}(f, b) = b$, which means $\pi(f, b) = b = \pi(g, b) - 1$. In this case, all the elements of \vec{i} must be smaller than b . We have

$$\delta(f) = \delta(g) = \delta(a \cdot \vec{i}; \vec{j}).$$

Therefore, $\Psi(g) = \Psi(f) + 1$.

Case 2.2. $\text{Inv}(f, b) > b$, which means $\pi(f, b) = b + 1 = \pi(g, b)$. In this case, there must be $c > b$ in \vec{i} . We have

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}; b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c]; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}; a \cdot \vec{j}) = \delta(b \cdot \vec{i}[a/c]; \vec{j}).\end{aligned}$$

Lemma 14 implies $|\delta(g) - \delta(f)| = 1$. Therefore, $\Psi(g) = \Psi(f) + 1$. \square

Lemma 17. Suppose that g is obtained from f by swapping a non-negative token and a negative one. Then $|\Psi(g) - \Psi(f)| = 1$.

Proof. **Case 1.** The swap takes place on non-negative vertices. Let

$$\begin{aligned}f &= (\vec{i}; \vec{j}_1 \cdot \langle a, b \rangle \cdot \vec{j}_2), \\ g &= (\vec{i}; \vec{j}_1 \cdot \langle b, a \rangle \cdot \vec{j}_2),\end{aligned}$$

where $a < 0 \leq b$. Obviously, $\mu(f) = \mu(g)$, $\text{Inv}(g, b) = \text{Inv}(f, b) + 1$ and $\text{Inv}(g, k) = \text{Inv}(f, k)$ for any other $k \geq 0$. By Lemma 11, there exists \vec{i}' consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\begin{aligned}\delta(f) &= \delta(\vec{i}'; \langle a, b \rangle \cdot \vec{j}_2) + \alpha, \\ \delta(g) &= \delta(\vec{i}'; \langle b, a \rangle \cdot \vec{j}_2) + \alpha\end{aligned}$$

for some $\alpha \leq 0$.

Case 1.1. $\text{Inv}(f, b) < b$ and $\text{Inv}(g, b) < b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. Corollary 3 applies to both f and g and we obtain $\delta(g) = \delta(f)$ and $\Psi(g) = \Psi(f) + 1$.

Case 1.2. $\text{Inv}(f, b) = b$ and $\text{Inv}(g, b) = b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. It is enough to show $\delta(f) = \delta(g)$. Since \vec{j}_2 contains b tokens smaller than b , $\vec{i} \cdot \vec{j}_1$ and \vec{i}' contain exactly $m - 1$ tokens smaller than b . Let $c = \max(\vec{i}')$, which is the unique element of \vec{i}' bigger than b . Let $(\vec{i}''; d) = \gamma(\vec{i}'; a)$.

Suppose $c = d$. We have $\gamma(\vec{i}'[b/d]; a) = (\vec{i}''; b)$, where b is the biggest in $\vec{i}'[b/d]$. Thus

$$\begin{aligned}\delta(f) &= \delta(\vec{i}''; b \cdot \vec{j}_2) - 1 + \alpha = \delta(\vec{i}''; \vec{j}_2) - 1 + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/d]; a \cdot \vec{j}_2) + \alpha = \delta(\vec{i}''; \vec{j}_2) - 1 + \alpha\end{aligned}$$

by Lemma 10.

If $c > d$, we have $\gamma(\vec{i}'[b/c]; a) = (\vec{i}''[b/c]; d)$ and

$$\begin{aligned}\delta(f) &= \delta(\vec{i}''; \langle d, b \rangle \cdot \vec{j}) + \alpha = \delta(\vec{i}''[d/c]; b \cdot \vec{j}) + \alpha = \delta(\vec{i}''[d/c]; \vec{j}) + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c]; a \cdot \vec{j}) + \alpha = \delta(\vec{i}''[b/c]; d \cdot \vec{j}) + \alpha = \delta(\vec{i}''[d/c]; \vec{j}) + \alpha\end{aligned}$$

by Lemma 10.

Case 1.3. $\text{Inv}(f, b) > b$ and $\text{Inv}(g, b) > b + 1$. In this case, we have $\pi(g) = \pi(f)$. It is enough to show $|\delta(g) - \delta(f)| = 1$. Since \vec{j}_2 contains at least $b + 1$ tokens smaller than b , $\vec{i} \cdot \vec{j}_1$ contains at most $m - 2$ tokens smaller than b . Hence \vec{i}' contains at least 2 tokens bigger than b . Let c_1 and c_2 be the biggest and second biggest in \vec{i}' , respectively. Let $(\vec{i}'', d) = \gamma(\vec{i}'; a)$.

If $d = c_1$, then by Lemma 10,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}''; b \cdot \vec{j}) - 1 + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c_1]; a \cdot \vec{j}) + \alpha = \delta(\vec{i}''; b \cdot \vec{j}) + \alpha.\end{aligned}$$

If $d = c_2$, then by Lemma 10,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}''; \langle c_2, b \rangle \cdot \vec{j}) + \alpha = \delta(\vec{i}''[c_2/c_1]; b \cdot \vec{j}) + \alpha = \delta(\vec{i}''[b/c_1]; \vec{j}) + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c_1]; a \cdot \vec{j}) + \alpha = \delta(\vec{i}''[b/c_1]; \vec{j}) - 1 + \alpha.\end{aligned}$$

If $d < c_2$, then by Lemma 10,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}''; \langle d, b \rangle \cdot \vec{j}) + \alpha = \delta(\vec{i}''[d/c_1][b/c_2]; \vec{j}) + \alpha, \\ \delta(g) &= \delta(\vec{i}'[b/c_1]; a \cdot \vec{j}) + \alpha = \delta(\vec{i}''[b/c_1]; d \cdot \vec{j}) + \alpha = \delta(\vec{i}''[b/c_1][d/c_2]; \vec{j}) + \alpha.\end{aligned}$$

Lemma 14 implies $|\delta(g) - \delta(f)| = |\delta(\vec{i}''[b/c_1][d/c_2]) - \delta(\vec{i}''[d/c_1][b/c_2])| = 1$.

Case 2. The swap takes place on 0 and a negative vertex. Without loss of generality we may assume

$$\begin{aligned}f &= (a \cdot \vec{i}; b \cdot \vec{j}), \\ g &= (b \cdot \vec{i}; a \cdot \vec{j}),\end{aligned}$$

where $a < 0 \leq b$. For the case where $a = -1$, we have already proved that $\Psi(g) = \Psi(f) + 1$ in Lemma 13. Hereafter we assume that $a \neq -1$. Clearly $\pi(g, i) = \pi(f, i)$ for all $i \in \{0, \dots, n\} - \{b\}$ and $\pi(g, b) = b + 1$. $\pi(f, b) = b$ if and only if every token in \vec{i} is smaller than b . Let $(\vec{i}', d) = \gamma(b \cdot \vec{i}; a)$.

Case 2.1. Suppose $\pi(f, b) = b$, in which case $\pi(g) = \pi(f) + 1$.

If $d = b$, there is $k \geq 1$ such that $g^i(a) < -1$ for $i \in \{0, \dots, k - 1\}$ and $g^k(a) = -1$. Since $f(i) = g(i)$ for $i < -1$ and $f(-1) = a$, we have $[a]_f \in \Delta_f$. Hence $\Delta_f = \Delta_g \cup \{[a]_f\}$ and thus $\mu(g) = \mu(f) - 1$. We have

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}; b \cdot \vec{j}) = \delta(a \cdot \vec{i}; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}; a \cdot \vec{j}) = \delta(\vec{i}'; \vec{j}) - 1.\end{aligned}$$

Since \vec{i}' is the a -resolution of $a \cdot \vec{i}$, by Lemma 15, we have $\delta(\vec{i}'; \vec{j}) = \delta(a \cdot \vec{i}; \vec{j})$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d < b$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$ and \vec{i}' has the form $b \cdot \vec{i}''$.

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}''; d \cdot \vec{j}) = \delta(d \cdot \vec{i}''; \vec{j}).\end{aligned}$$

Since $d \cdot \vec{i}''$ is the a -resolution of $a \cdot \vec{i}$, by Lemma 15, we have $\delta(a \cdot \vec{i}; \vec{j}) = \delta(d \cdot \vec{i}''; \vec{j})$ and thus $\Psi(g) = \Psi(f) + 1$.

Case 2.2. Suppose $\pi(f, b) = b + 1$, in which case $\pi(g) = \pi(f)$. Since there are at least $b + 1$ tokens in \vec{j} smaller than b , there are at most $m - 1$ tokens smaller than b in \vec{i} . Let $c = \max(\vec{i})$, which is therefore bigger than b .

If $d = c$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$ and

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}[b/c]; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}; a \cdot \vec{j}) = \delta(b \cdot \vec{i}'; \vec{j}) - 1.\end{aligned}$$

Since $b \cdot \vec{i}'$ is the a -resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(g) = \delta(f) - 1$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d = b$, $[a]_f \in \Delta_f$ and $[b]_g \notin \Delta_g$ by the same reason as in Case 2.1. In this case, $\mu(g) = \mu(f) - 1$ and

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}; b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c]; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}; a \cdot \vec{j}) = \delta(\vec{i}'; b \cdot \vec{j}) = \delta(\vec{i}'[b/c]; \vec{j}).\end{aligned}$$

Since $\vec{i}'[b/c]$ is the a -resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(f) = \delta(g)$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d \notin \{b, c\}$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$. There is \vec{i}'' such that $\vec{i}' = b \cdot \vec{i}''$. Let h be obtained from g by exchanging the tokens b and d . Then $|\delta(h) - \delta(g)| = 1$ and

$$\begin{aligned}\delta(f) &= \delta(a \cdot \vec{i}; b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c]; \vec{j}), \\ \delta(g) &= \delta(b \cdot \vec{i}; a \cdot \vec{j}) = \delta(b \cdot \vec{i}''; d \cdot \vec{j}) = \delta(b \cdot \vec{i}''[d/c]; \vec{j}), \\ \delta(h) &= \delta(d \cdot (\vec{i}[b/d]); a \cdot \vec{j}) = \delta(d \cdot \vec{i}''; b \cdot \vec{j}) = \delta(d \cdot \vec{i}''[b/c]; \vec{j}).\end{aligned}$$

Since $d \cdot \vec{i}''[b/c]$ is the a -resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(f) = \delta(h)$ and thus $|\delta(g) - \delta(f)| = 1$. $|\Psi(g) - \Psi(f)| = 1$. \square

Lemma 18. Suppose that g is obtained from f by swapping negative tokens. Then $|\Psi(g) - \Psi(f)| = 1$.

Proof. Clearly $\pi(f) = \pi(g)$.

Case 1. The swap takes place on non-negative vertices. Clearly $\mu(f) = \mu(g)$. It is enough to show $|\delta(g) - \delta(f)| = 1$. We may assume by Lemma 11

$$\begin{aligned}\delta(f) &= \delta(\vec{i}; \langle a, b \rangle \cdot \vec{j}), \\ \delta(g) &= \delta(\vec{i}; \langle b, a \rangle \cdot \vec{j}),\end{aligned}$$

where $a, b < 0$. Let $\gamma(\vec{i}; a) = (\vec{i}_a, a')$ and $\gamma(\vec{i}; b) = (\vec{i}_b, b')$. It is easy to see that there is $\vec{i}_{a,b}$ such that $\gamma(\vec{i}_a; b) = (\vec{i}_{a,b}, b')$ and $\gamma(\vec{i}_b; a) = (\vec{i}_{a,b}, a')$. Without loss of generality we assume $0 \leq a' < b'$. Let c_1 and c_2 be the biggest and the second biggest in \vec{i} .

Case 1.1. $b' = c_1$. By Lemma 10,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_a; \langle a', b \rangle \cdot \vec{j}) = \delta(\vec{i}_a[a'/b']; b \cdot \vec{j}) = \delta(\vec{i}_{a,b}; \langle a' \rangle \cdot \vec{j}) - [a' = c_2], \\ \delta(g) &= \delta(\vec{i}_b; a \cdot \vec{j}) - 1 = \delta(\vec{i}_{a,b}; \langle a' \rangle \cdot \vec{j}) - 1 - [a' = c_2],\end{aligned}$$

where $[a' = c_2] = 1$ if $a' = c_2$ and $[a' = c_2] = 0$ otherwise. Therefore, $|\delta(f) - \delta(g)| = 1$.

Case 1.2. $b' = c_2$.

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_a; \langle a', b \rangle \cdot \vec{j}) = \delta(\vec{i}_a[a'/c_1]; b \cdot \vec{j}) = \delta(\vec{i}_{a,b}[a'/c_1]; \vec{j}) - 1, \\ \delta(g) &= \delta(\vec{i}_b; \langle b', a \rangle \cdot \vec{j}) = \delta(\vec{i}_b[b'/c_1]; a \cdot \vec{j}) = \delta(\vec{i}_{a,b}[b'/c_1]; \langle a' \rangle \cdot \vec{j}) = \delta(\vec{i}_{a,b}[a'/c_1]; \vec{j}).\end{aligned}$$

Case 1.3. $b' < c_2$.

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_{a,b}[a'/c_1][b'/c_2]; \vec{j}), \\ \delta(g) &= \delta(\vec{i}_b; \langle b', a \rangle \cdot \vec{j}) = \delta(\vec{i}_b[b'/c_1]; a \cdot \vec{j}) = \delta(\vec{i}_{a,b}[b'/c_1]; \langle a' \rangle \cdot \vec{j}) = \delta(\vec{i}_{a,b}[b'/c_1][a'/c_2]; \vec{j}).\end{aligned}$$

By Lemma 14, $|\delta(g) - \delta(f)| = 1$.

Case 2. The swap takes place between a negative vertex and 0. Without loss of generality we may assume

$$\begin{aligned}f &= (a \cdot \vec{i}; b \cdot \vec{j}), \\ g &= (b \cdot \vec{i}; a \cdot \vec{j}),\end{aligned}$$

where $a, b < 0$. For the case where $a = -1$ or $b = -1$, we have already proved that $|\Psi(g) - \Psi(f)| = 1$ in Lemma 13. So we assume $a, b \neq -1$. Let $(\vec{i}_b; b') = \gamma(a \cdot \vec{i}; b)$. There are negative tokens $b_0, \dots, b_k < 0$ in $a \cdot \vec{i}$ such that $b_i = f^i(b) < 0$ for all $i \leq k$ and $f(b_k) = b' \geq 0$. Similarly for $(\vec{i}_a; a') = \gamma(b \cdot \vec{i}; a)$, there are negative tokens $a_0, \dots, a_l < 0$ in $b \cdot \vec{i}$ such that $a_i = g^i(a) < 0$ for all $i \leq l$ and $g(a_l) = a' \geq 0$. Let θ_a and θ_b be replacements $[a_0/a_1, \dots, a_{l-1}/a_l, a_l/a']$ and $[b_0/b_1, \dots, b_{k-1}/b_k, b_k/b']$, respectively. Then $\vec{i}_a = (b \cdot \vec{i})\theta_a$ and $\vec{i}_b = (a \cdot \vec{i})\theta_b$.

Case 2.1. Suppose that the sequence $\langle a_0, \dots, a_l \rangle$ contains -1 . By $g(-1) = b$, we have

$$\langle a_0, \dots, a_l, a' \rangle = \langle a_0, \dots, a_{l-k-2}, -1, b_0, \dots, b_k, b' \rangle,$$

where $a_{k-l-1} = -1$, $a_{k-l} = b_0$ and $a' = b'$. Since $f^{l-k-1}(a) = g^{l-k-1}(a) = -1$ and $f(-1) = a$, we have $[a]_f = \{a_0, \dots, a_{l-k-1}\} \in \Delta_f$. On the other hand, $g^k(b) = f^k(b) = b' \geq 0$ means that $[b]_f \notin \Delta_f$ and $[b]_g = [a]_g \notin \Delta_g$. Therefore, $\mu(f) = \mu(g) + 1$. Observing that

$$\begin{aligned}\vec{i}_b &= (a \cdot \vec{i})\theta_b, \\ \vec{i}_a &= (b \cdot \vec{i})[a_0/a_1, \dots, a_{l-k-1}/a_{l-k}]\theta_b \\ &= (a \cdot \vec{i})[b_0/a_0][a_0/a_1, \dots, a_{l-k-1}/a_{l-k}]\theta_b \\ &= (a \cdot \vec{i})[a_{l-k-1}/a_0, a_0/a_1, \dots, a_{l-k-2}/a_{l-k-1}]\theta_b \\ &= (a \cdot \vec{i})\theta_b[a_0/a_1, \dots, a_{l-k-2}/a_{l-k-1}, a_{l-k-1}/a_0] \\ &= \vec{i}_b[a_0/a_1, \dots, a_{l-k-2}/a_{l-k-1}, a_{l-k-1}/a_0],\end{aligned}$$

we see that \vec{i}_a is the a_0 -resolution of \vec{i}_b . Therefore, by Lemmas 10 and 15,

$$\delta(f) = \delta(\vec{i}_b; b' \cdot \vec{j}) - d = \delta(\vec{i}_a; b' \cdot \vec{j}) - d = \delta(g),$$

where $d = 1$ if $b' = \max(\vec{i})$ and $b = 0$ otherwise. All in all, we have $|\Psi(f) - \Psi(g)| = 1$.

The case where $\langle b_0, \dots, b_k \rangle$ contains -1 can be treated in the same way.

Case 2.2. Suppose that -1 occurs neither in $\langle a_0, \dots, a_l \rangle$ nor $\langle b_0, \dots, b_k \rangle$. It is easy to see that the two sequences $\langle b_0, \dots, b_k, b' \rangle$ and $\langle a_0, \dots, a_l, a' \rangle$ have no common elements. Hence $[a_0]_f \notin \Delta_f$ and $[b_0]_g \notin \Delta_g$. We obtain $\mu(f) = \mu(g)$. Without loss of generality, we may assume $a' < b'$. Let $h = f[a'/b', b'/a']$ be obtained from f by exchanging the positions of the tokens a' and b' . Since Lemma 14 ensures $|\delta(f) - \delta(h)| = 1$, it is enough to show $\delta(g) = \delta(h)$. By Lemma 10 and the fact $a' < b' \leq \max(\vec{i})$,

$$\delta(g) = \delta(\vec{i}_a; a' \cdot \vec{j}) = \delta((b_0 \cdot \vec{i})\theta_a; a' \cdot \vec{j})$$

The b_0 -resolution of \vec{i}_a is

$$(b_0 \cdot \vec{i})\theta_a[b_0/b_1, \dots, b_{k-1}/b_k, b_k/b', b'/b_0] = (b' \cdot \vec{i})\theta_a\theta_b.$$

On the other hand,

$$\begin{aligned} \delta(h) &= \delta(f[a'/b', b'/a']) = \delta((a_0 \cdot \vec{i})[a'/b', b'/a'] [b_0/b_1, \dots, b_{k-1}/b_k, b_k/a']; a' \cdot \vec{j}) \\ &= \delta((a_0 \cdot \vec{i})[b_0/b_1, \dots, b_{k-1}/b_k, b_k/b', b'/a']; a' \cdot \vec{j}) = \delta((a_0 \cdot \vec{i})\theta_b[b'/a']; a' \cdot \vec{j}). \end{aligned}$$

The a_0 -resolution of $(a_0 \cdot \vec{i})\theta_b[b'/a']$ is given as

$$\begin{aligned} &(a_0 \cdot \vec{i})\theta_b[b'/a'] [a_0/a_1, \dots, a_{l-1}/a_l, a_l/b', b'/a_0] \\ &= (a_0 \cdot \vec{i})\theta_b[a_l/a'] [a_0/a_1, \dots, a_{l-1}/a_l, b'/a_0] \\ &= (b' \cdot \vec{i})\theta_b[a_0/a_1, \dots, a_{l-1}/a_l, a_l/a'] \\ &= (b' \cdot \vec{i})\theta_b\theta_a = (b' \cdot \vec{i})\theta_a\theta_b, \end{aligned}$$

since θ_a and θ_b are independent. Therefore, $\delta(g) = \delta(h)$ by Lemma 15. \square

Theorem 12. *The token swapping problem on star-path graphs can be solved in polynomial time.*

Proof. By Lemmas 16, 17, 18 and 13, the number of swaps needed is exactly $\Psi(f)$. Obviously Ψ is computable in polynomial time. \square

C Proof that the PPN-Separable 3SAT Is NP-hard

We show the NP-hardness of the Sep-SAT by a reduction from the (usual) 3SAT [3]. For a given CNF F on X , we may without loss of generality assume that for each $x \in X$, the positive literal x and the negative one $\neg x$ occur exactly

the same number of times in F . Otherwise, if x occurs k more times than $\neg x$ does, we add clauses $\{\neg x, y_i, \neg y_i\}$ to F for all $i \in \{1, \dots, k\}$ where y_i are new Boolean variables. Now, for a given CNF F on $X = \{x_1, \dots, x_m\}$ such that the positive and negative literals x_i and $\neg x_i$ occur exactly the same number of times for each Boolean variable $x_i \in X$, we construct $F' = F_1 \cup F_2 \cup F_3$ on X' such that

- F is satisfiable if and only if F' is satisfiable,
- each positive literal x_i occurs just once in each of F_1 and F_2 ,
- each negative literal $\neg x_i$ occurs just once in F_3 .

Let n_i be the number of occurrences of the positive literal x_i in F (thus of the negative literal $\neg x_i$) for each $x_i \in X$.

1. Let $X' = \{x_{i,j}, \bar{x}_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n_i\}$.
2. Let F_1 be obtained from F by replacing the j -th occurrence of the positive literal x_i with $x_{i,j}$, and the j -th occurrence of the negative literal $\neg x_i$ with $\bar{x}_{i,j}$ for $j \in \{1, \dots, n_i\}$.
3. Let $F_2 = \{\{x_{i,j}, \bar{x}_{i,j}\} \mid 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\}$.
4. Let $F_3 = \{\{\neg x_{i,j}, \neg \bar{x}_{i,j+1}\} \mid 1 \leq i \leq m \text{ and } 1 \leq j < n_i\} \cup \{\{\neg x_{i,n_i}, \neg \bar{x}_{i,1}\} \mid 1 \leq i \leq m\}$.

Clearly F' is an instance of the Sep-SAT. If a map $\phi : X \rightarrow \{0, 1\}$ satisfies F , then $\phi' : X' \rightarrow \{0, 1\}$ satisfies F' where $\phi'(x_{i,j}) = 1 - \phi'(\bar{x}_{i,j}) = \phi(x_i)$ for each i and j . Conversely, suppose that F' is satisfied by $\phi' : X' \rightarrow \{0, 1\}$. The clauses of F_2 and F_3 ensure that $\phi'(x_{i,j}) = 1 - \phi'(\bar{x}_{i,j}) = \phi'(x_{i,1})$ for all $j \in \{1, \dots, n_i\}$. Then it is now clear that ϕ defined by $\phi(x_i) = \phi'(x_{i,1})$ satisfies F .